Handbuch

CPU 928

Band 1
Hardware und Programmierung

Band 2
Integrierte Funktionen und Kommunikation

Best.-Nr.: HB92D
Rev. 00/27

Änderungen im Sinne des technischen Fortschritts vorbehalten

Handbuch

CPU 928 - Band 1

Hardware und Programmierung

Rev. 00/27

Änderungen im Sinne des technischen Fortschritts vorbehalten

Die Angaben in diesem Handbuch erfolgen ohne Gewähr. Änderungen des Inhalts können jederzeit ohne Vorankündigung erfolgen. Die in VIPA-Handbüchern beschriebene Software unterliegt den Bedingungen eines allgemeinen Lizenzvertrages. Sie haben eine Kopierlizenz, die Sie berechtigt, die Software innerhalb Ihres Unternehmens zu nutzen.

Zuwiderhandlungen verpflichten zu Schadenersatz.

© Copyright 2000 VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH
Ohmstraße 4, D-91074 Herzogenaurach

Tel.: +49 (91 32) 744-0

Fax.: +49 (91 32) 744-144

EMail: info@vipa.de

http://www.vipa.de

Hotline: +49 (91 32) 744-114

Alle Rechte vorbehalten

Haftungsausschluss

Der Inhalt dieses Handbuchs wurde auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden regelmäßig überprüft und erforderliche Korrekturen sind in den nachfolgenden Revisionen enthalten.

Für Verbesserungsvorschläge sind wir dankbar.

Warenzeichen

VIPA®	ist eingetragenes Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH
SIMATIC®	ist eingetragenes Warenzeichen der Siemens AG.
STEP® 5	ist eingetragenes Warenzeichen der Siemens AG.

Alle ansonsten im Text genannten Warenzeichen sind Warenzeichen der jeweiligen Inhaber und werden als geschützt anerkannt.

Über dieses Handbuch

Die Beschreibung für die CPU 928 gliedert sich in zwei Bände, wobei jeder Band das gesamte Inhaltsverzeichnis und den gesamten Index beinhaltet.

Die Bände sind wie folgt gegliedert:

Band 1:	Teil 1 ... Teil 11	Hardware und Programmierung
Band 2:	Teil 12 ... Teil 15	Integrierte Funktionen und Kommunikation

Gesamtübersicht:

Teil 1 Grundlagen....1-1
Teil 2 Hardwarebeschreibung....2-1
Teil 3 Anwenderprogramm....3-1
Teil 4 Programmbearbeitung....4-1
Teil 5 Betriebszustände....5-1
Teil 6 Unterbrechungs- und Fehlerbehandlung....6-1
Teil 7 Erweiterter Datenbaustein DX 0....7-1
Teil 8 Speicherbelegung und -organisation....8-1
Teil 9 Speicherzugriff über absolute Adressen....9-1
Teil 10 Mehrprozessorbetrieb und -kommunikation....10-1
Teil 11 PG-Schnittstellen und -Funktionen....11-1
Teil 12 Integrierte Sonderfunktionen....12-1
Teil 13 Integrierte Hantierungsbausteine....13-1
Teil 14 Rechnerkopplung RK512....14-1
Teil 15 Datenübertragung 3964(R)....15-1
A Anhang....A-1

Gesamtinhaltsverzeichnis

Benutzerhinweise 1
Sicherheitshinweise 2
Teil 1 Grundlagen....1-1
Sicherheitshinweis für den Benutzer....1-2
Allgemeines....1-4
Anwendungsbereiche....1-5
Leistungsmerkmale....1-5
Arbeitsweise einer CPU....1-6
Programme einer CPU....1-6
Operanden der CPU 928....1-7
Adressierungsarten....1-9
Vorgehensweise bei der Programmierung....1-10
Unterschiede zu Siemens CPUs....1-12
Lieferumfang und Gerätevarianten....1-18
Teil 2 Hardwarebeschreibung....2-1
Aufbau CPU 928....2-2
Bedien- und Anzeigeelemente....2-4
Leuchtdioden....2-7
Schalter....2-9
Serielle Schnittstellen....2-10
Speicher-Module....2-15
Inbetriebnahme....2-16
Anlaufverhalten....2-17
Technische Daten....2-18
Teil 3 Anwenderprogramm....3-1
Programmiersprache....3-2
Programm-, Organisations- und Schrittbausteine....3-13
Funktionsbausteine....3-19
Datenbausteine....3-29
Teil 4 Programmbearbeitung....4-1
Prinzip der Programmbearbeitung....4-2
Programmorganisation....4-3
Speicherung von Programm- und Datenbausteinen....4-8
Bearbeitung des Anwenderprogramms....4-10
Operationen mit Beispielen....4-13
Teil 5 Betriebszustände....5-1
Einführung und Übersicht....5-2
Programmbearbeitungsebenen....5-5
Betriebszustand STOP....5-8
Betriebszustand ANLAUF....5-11
Betriebszustand RUN....5-15
Teil 6 Unterbrechungs- und Fehlerbehandlung....6-1
Häufige Fehler im Anwenderprogramm....6-2
Fehlerinformationen....6-3
Steuerbits und Unterbrechungsstack....6-7

Fehlerbehandlung über Organisationsbausteine....6-21
Fehler im ANLAUF....6-24
Fehler im RUN und im ANLAUF....6-30
Teil 7 Erweiterter Datenbaustein DX 0....7-1
Anwendung....7-2
Aufbau des DX 0....7-3
Parameter für DX 0....7-5
Parametrierbeispiele....7-9
Teil 8 Speicherbelegung und -organisation....8-1
Struktur des Speichers CPU 928....8-2
Adressbereiche für die Peripherie und deren Programmierung....8-4
Organisation des Anwenderspeichers CPU 928....8-5
Teil 9 Speicherzugriff über absolute Adressen....9-1
Einführung....9-2
Speicherzugriffe über Adresse in AKKU 1....9-6
Speicherblöcke transferieren....9-15
Operationen mit dem Basisadressregister (BR-Register)....9-22
Teil 10 Mehrprozessorbetrieb und -kommunikation....10-1
Teil 11 PG-Schnittstellen und -Funktionen....11-1
Übersicht....11-2
PG-Funktionen....11-3
Tätigkeiten an Kontrollpunkten....11-16
Serielle Kopplung....11-17
Parallelbetrieb von zwei seriellen PG-Schnittstellen....11-17
Teil 12 Integrierte Sonderfunktionen....12-1
OB 110: Zugriff auf das Anzeigenbyte....12-5
OB 111: AKKU 1, 2, 3 und 4 löschen....12-7
OB 112/113: AKKU-Roll-Up/AKKU-Roll-Down....12-8
OB 120: "Alarme gemeinsam sperren" ein-/ausschalten....12-10
OB 121: "Weckalarme einzeln sperren" ein-/ausschalten....12-13
OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten....12-15
OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten....12-18
OB 134, OB 135, OB 136 und OB 139....12-20
OB 150: Systemzeit stellen/lesen....12-22
OB 151: Uhrzeitgesteuerten Weckalarm stellen/lesen....12-29
OB 152: Zyklusstatistik....12-36
OB 160 bis 163: Zählschleifen....12-43
OB 170: Bausteinstack (BSTACK) lesen....12-45
OB 180: Variabler Datenbaustein-Zugriff....12-50
OB 181: Datenbausteine (DB/DX) testen....12-54
OB 182: Datenbereich kopieren....12-56
OB 186: Speicher komprimieren....12-60
OB 190/192: Merker in Datenbaustein übertragen....12-62
OB 191/193: Datenblöcke in Merkerbereich übertragen....12-65
OB 216 bis 218: Kachelzugriffe....12-70
OB 216: Schreiben auf eine Kachel....12-72
OB 217: Lesen aus einer Kachel....12-74
OB 218: Belegung einer Kachel....12-76
OB 220: Vorzeichenerweiterung....12-78

OB 221: Zyklusüberwachung einstellen 12-79
OB 222: Zyklusüberwachungszeit neu starten 12-80
OB 228: Statusinformation einer Programmebene lesen 12-81
OB 230 bis 237: Funktionen für Standard-FBs 12-83
OB 250/251: Regelung /PID-Algorithmus 12-84
OB 250: PID-Algorithmus initialisieren 12-91
OB 251: PID-Algorithmus bearbeiten 12-92
OB 254/255: Einen Datenbaustein verschieben/duplizieren....12-98
Teil 13 Integrierte Hantierungsbausteine 13-1
Einführung 13-2
Die Parameter der Hantierungsbausteine 13-3
Parameterbeschreibung....13-4
Parametrierung von SSNR, A-Nr, ANZW und BLGR 13-6
Indirekte Parametrierung von Quell- und Zielangaben....13-9
Aufbau des Anzeigenwortes 13-10
Arbeiten mit dem Anzeigewort 13-12
Das Längen - Wort 13-15
Aufbau Parametrierfehlerbyte (PAFE) 13-16
Einstellbare Blockgrößen....13-16
Tabelle über die möglichen QTYP/ZTYP - Parameter 13-17
Der SEND - Baustein FB 120....13-19
Der RECEIVE-Baustein FB 121....13-22
Der FETCH-Baustein FB 122....13-25
Der CONTROL-Baustein FB 123....13-26
Der RESET-Baustein FB 124....13-27
Der SYCHRON-Baustein FB 125....13-28
SEND-ALL 126 13-29
RECEIVE-ALL 127 13-30
Blockgröße 13-31
Analogwert-Anpassungsbausteine FB 250 und FB 251 13-32
Beispiel für eine Analogwertverarbeitung....13-37
Teil 14 Rechnerkopplung RK512....14-1
Einführung RK512 14-2
Parametrierung....14-10
Koordinierungsmerker (KM)....14-26
Inbetriebnahme....14-28
Betrieb 14-30
Datenaustausch....14-38
Telegrammbeschreibung 14-43
Teil 15 Datenübertragung 3964(R) 15-1
Einführung 3964 und 3964R 15-2
Parametrierung....15-9
Inbetriebnahme....15-22
Betrieb 15-24
Prozeduren....15-30
A Anhang A-1
Gesamtindex A-1

Benutzerhinweise

Zielsetzung und Inhalt

Band 1 ist Bestandteil des Handbuchs für die CPU-928-Baugruppen von VIPA. Es beschreibt die Hardware und die Programmierung der CPU 928B und CPU 928C deren Einsatzmöglichkeiten und Projektierung.

Die CPU 928-Baugruppen können Sie im VIPA AG-135U und in allen Automatisierungsgeräten der (AG-115U ... AG-155U) einsetzen.

Zielgruppe

Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik und in der SPS-Programmierung.

Aufbau des Handbuchs

Dieser Band ist gegliedert in 11 Kapitel. Jedes Kapitel beschreibt eine abgeschlossene Thematik und umfasst maximal 68 Seiten.

Orientierung im Dokument

Als Orientierungshilfe stehen im Handbuch zur Verfügung:

- Gesamt Inhaltsverzeichnis am Anfang des Handbuchs
- Übersicht der beschriebenen Themen am Anfang jedes Kapitels
- Stichwortverzeichnis (Index) am Ende des Handbuchs

Verfügbarkeit

Das Handbuch ist verfügbar in:

- gedruckter Form auf Papier
- in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)

Piktogramme Signalwörter

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

Gefahr!

Unmittelbar drohende oder mögliche Gefahr.

Personenschäden sind möglich.

Achtung!

Bei Nichtbefolgen sind Sachschäden möglich.

Hinweis!

Zusätzliche Informationen und nützliche Tips.

Sicherheitshinweise

Bestimmungs-gemäße Verwendung

Die CPU-Baugruppen sind konstruiert und gefertigt für

- Allgemeine Steuerungs- und Automatisierungsaufgaben
- den industriellen Einsatz
- den Betrieb innerhalb der in den technischen Daten spezifizierten Umgebungsbedingungen
- den Einbau in ein geschlossenes, HF-dichtes Metallgehäuse

Gefahr!

Das Gerät ist nicht zugelassen für den Einsatz

- in explosionsgefährdeten Umgebungen (EX-Zone)

Dokumentation

Handbuch zugänglich machen für alle Mitarbeiter in

- Projektierung
- Installation
- Inbetriebnahme
- Betrieb

Vor Inbetriebnahme und Betrieb der in diesem Handbuch beschriebenen Komponenten unbedingt beachten:

- Änderung am Automatisierungssystem nur im spannungslosen Zustand vornehmen!
- Anschluss und Änderung nur durch ausgebildetes Elektro-Fachpersonal
- Nationale Vorschriften und Richtlinien im jeweiligen Verwenderland beachten und einhalten (Installation, Schutzmaßnahmen, EMV ...)

Entsorgung

Zur Entsorgung der Baugruppen nationale Vorschriften beachten!

Teil 1 Grundlagen

Überblick

Im Rahmen dieser Einleitung folgen Hinweise im Umgang und Informationen über Einsatzbereiche und Anwendung der CPU928-Baugruppen.

Sie erhalten ferner einige Vorschläge, wie Sie beim Programmieren vorgehen können und erfahren, welche für die Programmierung wichtige Kenndaten die CPU 928 hat.

Eine Übersicht der Unterschiede und Verbesserungen zur Original CPU928 beenden die Einleitung.

Nachfolgend sind beschrieben:

- Sicherheitshinweise für den Benutzer
- Aufbau und Arbeitsweise der CPU 928
- Grundlagen der Programmierung
- Unterschiede und Verbesserungen gegenüber der SIEMENS CPU

Inhalt

Thema	Seite
Teil 1 Grundlagen	**1-1**
Sicherheitshinweis für den Benutzer	1-2
Allgemeines	1-4
Anwendungsbereiche	1-5
Leistungsmerkmale	1-5
Arbeitsweise einer CPU	1-6
Programme einer CPU	1-6
Operanden der CPU 928	1-7
Adressierungsarten	1-9
Vorgehensweise bei der Programmierung	1-10
Unterschiede zu Siemens CPUs	1-12
Lieferumfang und Gerätevarianten	1-18

Baugruppen-Übersicht

Typ	Bestellnummer	Beschreibung
CPU 928B	VIPA 928-3UB12	CPU 928B (EPROM)
CPU 928B	VIPA 928-3UB21	CPU 928B (Flash-ROM)
CPU 928C	VIPA 928-3UC11	CPU 928C (Flash-ROM)

Sicherheitshinweis für den Benutzer

Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Baugruppen sind mit hochintegrierten Bauelementen in MOS-Technik bestückt. Diese Bauelemente sind hoch empfindlich gegenüber Überspannungen, die z.B. bei elektrostatischer Entladung entstehen.

Zur Kennzeichnung dieser gefährdeten Baugruppen wird nachfolgendes Symbol verwendet:

Das Symbol befindet sich auf Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Baugruppen hin.

Elektrostatisch gefährdete Baugruppen können durch Energien und Spannungen zerstört werden, die weit unterhalb der Wahrnehmungsgrenze des Menschen liegen. Hantiert eine Person, die nicht elektrisch entladen ist, mit elektrostatisch gefährdeten Baugruppen, können diese Spannungen auftreten und zur Beschädigung von Bauelementen führen und so die Funktionsweise der Baugruppen beeinträchtigen oder die Baugruppe unbrauchbar machen. Auf diese Weise beschädigte Baugruppen werden in den wenigsten Fällen sofort als fehlerhaft erkannt. Der Fehler kann sich erst nach längerem Betrieb einstellen.

Durch statische Entladung beschädigte Bauelemente können bei Temperaturänderungen, Erschütterungen oder Lastwechseln zeitweilige Fehler zeigen.

Nur durch konsequente Anwendung von Schutzeinrichtungen und verantwortungsbewusste Beachtung der Handhabungsregeln lassen sich Funktionsstörungen und Ausfälle an elektrostatisch gefährdeten Baugruppen wirksam vermeiden.

Versenden von elektrostatisch gefährdeten Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung. Zusätzlich können Sie die für den Versand vorbereiteten Baugruppen mit einer leitfähigen Verpackung umhüllen. Leitfähige Verpackungen sind antistatische Folien oder metallisierte Kunststoffschachteln.

Achtung!

Auf manchen Baugruppen befindet sich eine Batterie. Beachten Sie beim Versenden in einer leitfähigen Verpackung, dass die Batterieanschlüsse nicht berührt oder kurzgeschlossen werden.

Messen und Ändern von elektrostatisch gefährdeten Baugruppen

Bei Messungen an elektrostatisch gefährdeten Baugruppen sind folgende Dinge zu beachten:

- Potentialfreie Messgeräte sind kurzzeitig zu entladen.
- Verwendete Messgeräte sind zu erden.

Bei Änderungen an elektrostatisch gefährdeten Baugruppen ist darauf zu achten, dass ein geerdeter Lötkolben verwendet wird.

Achtung!

Bei Arbeiten mit und an elektrostatisch gefährdeten Baugruppen ist auf ausreichende Erdung des Menschen und der Arbeitsmittel zu achten.

Allgemeines

Das Automatisierungsgerät 135U, 155U bzw. 188U gehört zur Familie der speicherprogrammierbaren Steuerungen. In Verbindung mit der CPU 928 erhalten Sie ein leistungsfähiges Gerät zur Prozessautomatisierung.

CPU 928B CPU 928B CPU 928C

928-3UB12 928-3UB21 928-3UC11

Dieses System können Sie sowohl für den Aufbau einfachster Steuerungen mit binären Signalen als auch zur Lösung umfangreicher Automatisierungsaufgaben einsetzen.

Die CPU 928B ist funktionsgleich zu der CPU 928B von Siemens.

Sie erhalten die 928B in zwei Ausführungen:

CPU 928B für EPROM-Modul 376

CPU 928B für Flash-Card 374

Eine Weiterentwicklung der CPU 928B stellt die CPU 928C dar.

Speichermedium ist die Flash-Card 374.

Die CPU 928C hat im Gegensatz zur CPU 928B einen größeren Anwenderspeicher. Sie besitzt 128 kByte Programmspeicher und zusätzlich 110 kByte Datenspeicher.

Die CPU 928C ist im Durchschnitt 25% schneller als die CPU 928B.

Hinweis!

Soweit nichts anderes erwähnt ist, beziehen sich die in diesem Handbuch gemachten Angaben auf die CPU 928-Familie von VIPA!

Im Weiteren wird nur noch von der CPU 928 gesprochen. Bei Funktionsabweichungen der einzelnen CPU 928-Typen wird darauf hingewiesen!

Anwendungsbereiche

Die CPU 928 können Sie im AG-135U/155U/188U und im Rack-135U von VIPA einsetzen.

Die Zentralbaugruppe CPU 928 ist für Anwendungen im mittleren Leistungsbereich ausgelegt.

Sie eignet sich besonders für:

- Aufgaben, die eine schnelle Bit- und Wortverarbeitung erfordern, d.h. schnelles Steuern und Regeln
- Aufgaben, die eine schnelle Kommunikation erfordern wie z.B. Anbindung an Bussysteme und Visualisierungen
- Regelungsaufgaben sowie deren Bedienung und Beobachtung
- Automatisierungsstrukturen, die eine direkte Kommunikation zu einem übergeordneten Leitrechner, einem anderen AG oder einem Drucker über eine Busverbindung oder Rechnerkopplung erfordern

Besonderheiten

Die Baugruppe verfügt über zwei Schnittstellen, wobei die 2. Schnittstelle (SI 2) über Hybridmodule frei konfigurierbar ist.

Über die 2. Schnittstelle SI 2 können Sie einfach zu handhabende Punkt zu Punkt Verbindungen realisieren.

Leistungsmerkmale

CPU-928-Baugruppe

- Programmkompatibilität durch Ablauf des gesamten Befehlsvorrats
- Anschluss für PG
- zweite frei konfigurierbare serielle Schnittstelle
- Schmale Bauweise
- Steckplatz für Speichermodule
- Hardware-Uhr

Kommunikation

- PG-Schnittstelle
- frei konfigurierbare serielle Schnittstelle (über Hybrid-Module frei konfigurierbar)

Arbeitsweise einer CPU

In einer CPU gibt es folgende Arbeitsweisen:

zyklische Bearbeitung

Die **zyklische** Bearbeitung stellt den Hauptanteil aller Vorgänge in der CPU. In einem endlosen Zyklus werden die gleichen Bearbeitungsfolgen wiederholt.

zeitgesteuerte Bearbeitung

Erfordern Prozesse in konstanten Zeitabschnitten Steuersignale, so können Sie neben dem zyklischen Ablauf **zeitgesteuert** bestimmte Aufgaben durchführen z.B. zeitunkritische Überwachungsfunktionen im Sekundenraster.

alarmgesteuerte Bearbeitung

Soll auf ein Prozesssignal besonders schnell reagiert werden, so ordnen Sie diesem einen **alarmgesteuerten** Bearbeitungsabschnitt zu. Ein Alarm kann in Ihrem Programm eine Bearbeitungsfolge aktivieren.

Bearbeitung nach Priorität

Die oben genannten Bearbeitungsarten werden von der CPU nach Wichtigkeitsgrad behandelt **(Priorität)**. Da auf ein Zeit- oder Alarmereignis schnell reagiert werden muss, unterbricht zur Bearbeitung dieser hochprioren Ereignisse die CPU die zyklische Bearbeitung, reagiert auf diese Ereignisse und setzt danach die zyklische Bearbeitung wieder fort. Die zyklische Bearbeitung hat daher die niedrigste Priorität.

Programme einer CPU

Das in jeder CPU vorhandene Programm unterteilt sich in:

Systemprogramm

Das Systemprogramm organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind.
Die nachfolgenden Kapitel außer Teil "Erweiterter Datenbaustein DX0" beschreiben das voreingestellte Systemverhalten bei Reaktionen auf Prozessereignisse oder Fehler. Sie können dieses Systemverhalten durch Parametrieren des Datenbausteins DX0 modifizieren .

Anwenderprogramm

Hier finden Sie alle Funktionen, die zur Bearbeitung einer spezifischen Steuerungsaufgabe erforderlich sind. Schnittstellen zum Systemprogramm stellen die Operationsbausteine zur Verfügung.

Operanden der CPU 928

Die CPU 928 stellt Ihnen für das Programmieren folgende Operandenbereiche zur Verfügung:

Prozessabbild und Peripherie

Auf das Prozessabbild der Aus- und Eingänge PAA/PAE kann Ihr Anwenderprogramm sehr schnell zugreifen. Sie haben Zugriff auf folgende Datentypen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

Sie können mit Ihrem Anwenderprogramm über den Rückwand-Bus direkt auf Peripheriebaugruppen zugreifen. Folgende Datentypen sind möglich:

- Bytes
- Wörter

Merker (M- und S-Merker)

Der Merkerbereich ist ein Speicherbereich, auf den Sie über Ihr Anwenderprogramm mit entsprechenden Operationen zugreifen können. Verwenden Sie den Merkerbereich für oft benötigte Arbeitsdaten. Sie können auf folgende Datentypen zugreifen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

Die CPU 928 enthält einen zusätzlichen Merkerbereich, den S-Merkerbereich. S-Merker lassen sich nicht als Aktualoperanden bei Funktionsbaustein-Aufrufen benutzen.
Voraussetzung für den Einsatz von S-Merkern ist das MC5-Programmierpaket von VIPA oder "S5-DOS/MT" von Siemens.

Zeiten und Zähler

Sie können mit Ihrem Anwendungsprogramm eine Zeitzelle mit einem Wert zwischen 10ms und 9990s laden. Sobald Ihr Anwenderprogramm eine Startoperation ausführt, wird dieser Zeitwert um ein durch Sie vorgegebenes Zeitraster dekrementiert, bis Null erreicht wird.

Für den Einsatz von Zählern können Sie Zählerzellen mit einem Anfangswert laden (max. 999) und diesen hinauf- bzw. herunterzählen.

Datenbausteine

Ein Datenbaustein enthält Konstanten bzw. Variablen im Byte-, Wort- oder Doppelwortformat. Mit Operanden können Sie immer auf den aktuellen Datenbaustein zugreifen.

Sie haben Zugriff auf folgende Datentypen:

- Einzelbits
- Bytes
- Wörter
- Doppelwörter

Adressierungsarten

Für den Zugriff auf Operanden- und Speicherbereiche können Sie zwei Adressierungsarten einsetzen:

Relative Adressierung

Der überwiegende Teil der Befehle adressiert eine Speicherzelle relativ zum Beginn eines Operandenbereichs. Wenn ausschließlich mit diesen Befehlen gearbeitet wird, sind Code- und Datenbereiche des Anwenderprogramms gegen ungewolltes Überschreiben geschützt. Zugleich ist das Anwenderprogramm unabhängig von der verwendeten CPU, sofern die CPU über einen entsprechenden Operandenbereich verfügt.

Absolute Adressierung

Es gibt auch Befehle, die mit absoluter Adressierung arbeiten. Mit diesen Befehlen können Sie auf den ganzen Speicherbereich zugreifen. Sie lassen sich aber nur in Funktionsbausteinen verwenden und sollten wegen der Gefahr von Datenzerstörung nur mit größter Vorsicht eingesetzt werden. Diese Befehle sind vom CPU-Typ abhängig.

Datenbausteine werden vom Systemprogramm in den Anwenderspeicher oder in das DB-RAM geladen. Bei der CPU 928 werden Datenbausteine nur im DB-RAM abgelegt. Ihre Lage ist abhängig vom jeweils verfügbaren Speicherplatz. Die Länge der einzelnen Datenbausteine kann unterschiedlich sein und wird beim Programmieren eines Datenbausteins festgelegt.

Der aktuelle Datenbaustein ist der Datenbaustein, dessen Anfangsadresse und Länge in speziellen Registern eingetragen ist. Dieser Eintrag erfolgt über eine spezielle Operation zum Aufrufen eines Datenbausteins. Das Anwenderprogramm kann, sofern sie keine Befehle mit absoluter Adressierung einsetzen, ausschießlich auf den aktuellen Datenbaustein zugreifen.

Folgende Datentypen sind möglich:

- Einzelbit
- Bytes
- Wörter
- Doppelwörter

Vorgehensweise bei der Programmierung

Ein Steuerungsprogramm lässt sich in drei Abschnitte einteilen:

- Technologisch Aufgabenstellung
- Programmentwurf
- Programmierung, Test und Inbetriebnahme.

technologische Aufgabenstellung

- Erstellen Sie ein grobes Blockdiagramm über die Steuerungsaufgaben Ihres Prozesses.
- Erstellen Sie eine Liste der für die Aufgabe benötigten Ein- und Ausgabesignale.
- Verfeinern Sie das Blockdiagramm, indem Sie den einzelnen Blöcken die Signale und evtl. erforderlichen Zeitbedingungen bzw. Zählerstände zuordnen.

Programm entwerfen

- Entwerfen Sie die Bearbeitungsarten (zyklisch bzw. zeitgesteuert) Ihres Programms und benennen Sie die hierzu verwendeten OBs.
- Teilen Sie die Bearbeitungsarten in technologische bzw. funktionale Blöcke ein.
- Prüfen Sie, ob Sie die Blöcke einem Programm- oder Funktionsbaustein zuordnen können und benennen Sie die zu verwendenden Bausteine (PB x, FB y usw.)
- Klären Sie den Speicherbedarf für Zähler, Zeiten, Daten- und Ergebnisspeicher.
- Legen Sie die Aufgaben für jeden vorgesehenen Codebaustein und die Daten für evtl. benötigte Merker und Datenbausteine fest.
- Zeichnen Sie für die Codebausteine Ablaufdiagramme.

Hinweis!

Der Zyklus muss ausreichend schnell ablaufen. Die Prozesszustände dürfen sich nicht schneller ändern als die CPU darauf reagieren kann. Ansonsten könnte der Prozess außer Kontrolle geraten. Berücksichtigen Sie als maximale Reaktionszeit die doppelte Zykluszeit.

Sie bestimmen die Zykluszeit durch die zyklische Bearbeitung des Systemprogramms und durch Art und Umfang des Anwenderprogramms.

Programmierung, Test und Inbetriebnahme

- Legen Sie die Darstellungsart für die Codebausteine fest (FUP, KOP oder AWL).
- Programmieren Sie die Code- und Datenbausteine.
- Nehmen Sie die Bausteine nacheinander in Betrieb.

Wenn Sie sicher sind, dass alle Codebausteine richtig ablaufen und alle Daten richtig berechnet und abgespeichert werden, können Sie Ihr gesamtes Programm in Betrieb nehmen.

Hinweis!

Bitte beachten Sie, dass Sie die Funktionsbausteine nur in der Darstellungsart AWL erstellen können.

Unterschiede zu Siemens CPUs

CPU 928B Version 3UB12

- Zwei PG-Schnittstellen, die beide gleichzeitig genutzt werden können, sofern unterschiedliche Funktionen angesprochen werden.
- Kein Mehrprozessorbetrieb.
- Kein Neustart mit Gedächtnis.
- Ca. doppelte Geschwindigkeit bei Wortbefehlen.
- Nach dem Urlöschen wird Ihr Programm vom Modul in den CPU-Speicher kopiert. Danach kann das Modul entfernt werden.
- Schreibschutz für Anwenderspeicher aktivieren/ deaktivieren mit BS138 Bit 0. Bausteine mit EPROM-Kennung können vom PG nicht überschrieben werden, jedoch vom Anwenderprogramm!
- Kopieren des Anwenderprogramms in das interne Flash-EPROM der CPU mit BS138 Bit 15.
  Wenn der Schreibschutz für den Anwenderspeicher aktiviert wird (BS138 mit Bit 0 = 1), kann durch zusätzliches Setzen des Bit 15 das Anwenderprogramm und DB-RAM im Flash-EPROM der CPU gesichert werden. Dies ermöglicht der CPU das Nachladen, falls ein Fehler im Anwenderspeicher nach NAU festgestellt wird.
  Das Nachladen erfolgt nur, wenn das Anwenderprogramm zwischenzeitlich nicht verändert wurde!
  Vorgehen:
  1. Stoppen Sie die CPU
  2. Lassen Sie sich den Inhalt der Adresse EA8Ah am PG ausgeben.
  3. Setzen Sie das Bit 0 und das Bit 15 (Bitmuster 8001h).
  4. Führen Sie anschließend eine NEUSTART durch.
- Kein Softwareschutz durch BS139.
- Speicher-Füllstandsanzeiger für den Anwenderspeicher in BS64, für das DB-RAM in BS65.
- Systemfehler
  Falls ein Systemfehler auftritt wird
  - der SPS-Prozessor gestoppt und alle Ausgänge (P,Q) werden gelöscht
  - im BS7 wird UAPRFE gesetzt (Urlöschen, kein Anlauf)
  - im BS8 wird FE5 gesetzt (Systemfehler)
  - im BS80 erfolgt eine nähere Spezifizierung der Ursache
  - die CPU führt einen Kaltstart durch

Systemfehler-Anzeige

		LEDs		
Fehlerursache	**BS80**	**QVZ**	**ADF**	**ZYK**
Hardware Watchdog	FE01h	O	O	X
CP-SPS Kommunikation	FE02h	O	X	O
Überwachung SPS-Prozessor	FE03h	O	X	X
Tasküberwachung CP-Prozesssor	FE04h	X	O	O
Nicht belegter Interrupt	FE05h	X	O	X
Speicherfehler	FE06h	X	X	O

O → LED aus; X → LED blinkt

Integrierte Sonder-OBs:

OB 110: Zugriff auf das Anzeigenbyte
OB 111: AKKU 1, 2, 3 und 4 löschen
OB 112/113: AKKU-Roll-Up / AKKU-Roll-Down
OB 120: "Alarme gemeinsam sperren" ein-/ausschalten
OB 121: "Weckalarme einzeln sperren" ein-/ausschalten
OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten
OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten
OB 134: *D
OB 135: /D
OB 136: MOD
OB 139: PUSH
OB 150: Systemzeit stellen/lesen
OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen
OB 152: Zyklusstatistik
OB 160 bis 163: Zählschleifen
OB 170: Bausteinstack (BSTACK) lesen
OB 180: Variabler Datenbaustein-Zugriff
OB 181: Datenbausteine (DB/DX) testen
OB 182: Datenbereich kopieren
OB 186: Speicher komprimieren
OB 190/OB 192: Merker in DB übertragen
OB 191/OB 193: Datenblöcke in Merkerbereich übertragen
OB 216: Schreiben auf eine Kachel
OB 217: Lesen aus einer Kachel
OB 218: Belegen einer Kachel
OB 220: Vorzeichenerweiterung
OB 221: Zyklusüberwachungszeit einstellen
OB 222: Zyklusüberwachungszeit neu starten
OB 228: Statusinformation einer Programmebene lesen
OB 230: HTB-Send
OB 231: HTB-Receive
OB 232: HTB-Fetch
OB 233: HTB-Control
OB 234: HTB-Reset
OB 235: HTB-Synchron
OB 236: HTB-Send ALL
OB 237: HTB-Receive ALL
OB 250: PID-Algorithmus initialisieren
OB 251: PID-Algorithmus bearbeiten
OB 254/ OB 255: Einen DB verschieben/ duplizieren

- Die Standardhantierungsbausteine unterstützen die S-Merkerbereiche.
- Datenübertragung mit den Prozeduren 3964 bzw. 3964R und RK512
  Die Datenübertragung erfolgt über die zweite serielle Schnittstelle (SI 2). Die Prozeduren können mit einer höheren Baudrate von bis zu 38400 Baud ausgeführt werden. Die Einstellung erfolgt im DX2 mit dem Wert 000Ah.

CPU 928B Version 3UB21

- Zwei PG-Schnittstellen, die beide gleichzeitig genutzt werden können, sofern unterschiedliche Funktionen angesprochen werden.
- Kein Mehrprozessorbetrieb.
- Kein Neustart mit Gedächtnis.
- Ca. doppelte Geschwindigkeit bei Wortbefehlen.
- Nach dem Urlöschen wird Ihr Programm vom PCMCIA-Modul in den CPU-Speicher kopiert. Danach können Sie das PCMCIA-Modul entfernen.
- Schreibschutz für Anwenderspeicher aktivieren/ deaktivieren mit BS138 Bit 0. Bausteine mit EPROM-Kennung können vom PG nicht überschrieben werden, jedoch vom Anwenderprogramm!
- Kopieren des Anwenderprogramms in das interne Flash-EPROM der CPU mit BS138 Bit 15.
  Wenn der Schreibschutz für den Anwenderspeicher aktiviert wird (BS138 Bit 0), kann durch zusätzliches Setzen des Bit 15 das Anwenderprogramm und DB-RAM im Flash-EPROM der CPU gesichert werden. Dies ermöglicht der CPU das Nachladen, falls ein Fehler im Anwenderspeicher nach NAU festgestellt wird.
  Das Nachladen erfolgt nur wenn das Anwenderprogramm zwischenzeitlich nicht verändert wurde!
  Vorgehen:
  1. Stoppen Sie die CPU
  2. Lassen Sie sich den Inhalt der Adresse EA8Ah am PG ausgeben.
  3. Setzen Sie das Bit 0 und das Bit 15 (Bitmuster 8001h).
  4. Führen Sie anschließend eine NEUSTART durch.
- Kein Softwareschutz durch BS139.
- Speicher-Füllstandsanzeiger für den Anwenderspeicher in BS64, für das DB-RAM in BS65
- Systemfehler
  Falls ein Systemfehler auftritt wird
  - der SPS-Prozessor gestoppt und alle Ausgänge (P,Q) werden gelöscht
  - im BS7 wird UAPRFE gesetzt (Urlöschen, kein Anlauf)
  - im BS8 wird FE5 gesetzt (Systemfehler)
  - im BS80 erfolgt eine nähere Spezifizierung der Ursache
  - die CPU führt einen Kaltstart durch

Systemfehler-Anzeige

Fehlerursache	BS80	LEDs		
		QVZ	ADF	ZYK
Hardware Watchdog	FE01h	O	O	X
CP-SPS Kommunikation	FE02h	O	X	O
Überwachung SPS-Prozessor	FE03h	O	X	X
Tasküberwachung CP-Prozesssor	FE04h	X	O	O
Nicht belegter Interrupt	FE05h	X	O	X
Speicherfehler	FE06h	X	X	O

O → LED aus; X → LED blinkt

Integrierte Sonder-OBs

OB 110: Zugriff auf das Anzeigenbyte
OB 111: AKKU 1, 2, 3 und 4 löschen
OB 112/113: AKKU-Roll-Up / AKKU-Roll-Down
OB 120: "Alarme gemeinsam sperren" ein-/ausschalten
OB 121: "Weckalarme einzeln sperren" ein-/ausschalten
OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten
OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten
OB 134: *D
OB 135: /D
OB 136: MOD
OB 139: PUSH
OB 150: Systemzeit stellen/lesen
OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen
OB 152: Zyklusstatistik
OB 160 bis 163: Zählschleifen
OB 170: Bausteinstack (BSTACK) lesen
OB 180: Variabler Datenbaustein-Zugriff
OB 181: Datenbausteine (DB/DX) testen
OB 182: Datenbereich kopieren
OB 186: Speicher komprimieren
OB 190/OB 192: Merker in DB übertragen
OB 191/OB 193: Datenblöcke in Merkerbereich übertragen
OB 216: Schreiben auf eine Kachel
OB 217: Lesen aus einer Kachel
OB 218: Belegen einer Kachel
OB 220: Vorzeichenerweiterung
OB 221: Zyklusüberwachungszeit einstellen
OB 222: Zyklusüberwachungszeit neu starten
OB 228: Statusinformation einer Programmebene lesen
OB 230: HTB-Send
OB 231: HTB-Receive
OB 232: HTB-Fetch
OB 233: HTB-Control
OB 234: HTB-Reset
OB 235: HTB-Synchron
OB 236: HTB-Send ALL
OB 237: HTB-Receive ALL
OB 250: PID-Algorithmus initialisieren
OB 251: PID-Algorithmus bearbeiten
OB 254/ OB 255: Einen DB verschieben/ duplizieren

- Die Standardhantierungsbausteine unterstützen die S-Merkerbereiche.
- Datenübertragung mit den Prozeduren 3964 bzw. 3964R und RK512
  Die Datenübertragung erfolgt über die zweite serielle Schnittstelle (SI 2). Die Prozeduren können mit einer höheren Baudrate von bis zu 38400 Baud ausgeführt werden. Die Einstellung erfolgt im DX2 im Datenwort 0 (relativ) mit dem Wert 000Ah.

CPU 928C Version 3UC11

- Die CPU besitzt zwei getrennte Speicherbereiche für Codebausteine (OB, FB etc.) und Datenbausteine.
- Zugriff auf den Anwenderspeicher (Codebausteine) nur über die Befehle LDI, TDI.
- Vom Programmiergerät ist der Anwenderspeicher nicht sichtbar.
- Zwei PG-Schnittstellen.
  Beide Schnittstellen können gleichzeitig genutzt werden, sofern unterschiedliche Funktionen angesprochen werden.
- Kein Mehrprozessorbetrieb.
- Kein Neustart mit Gedächtnis.
- Ca. doppelte Geschwindigkeit bei Wortbefehlen.
- Das Anwenderprogramm wird nach Urlöschen vom PCMCIA -Modul in den CPU-Speicher kopiert. Das PCMCIA -Modul kann nach dem Kopieren entfernt werden.
- Alternatives Laden von Datenbausteinen (BS144) wird nicht benötigt, da Datenbausteine immer im DB-RAM abgelegt werden.
- Schreibschutz für Anwenderspeicher aktivieren/ deaktivieren mit BS138 Bit 0.
  Bausteine mit EPROM-Kennung können vom PG nicht überschrieben werden, aber vom Anwenderprogramm!
- Kopieren des Anwenderprogramms in das interne Flash-EPROM der CPU mit BS138 ist, im Gegensatz zur VIPA CPU928B, nicht möglich.
- Kein Softwareschutz durch BS139.
- Speicher-Füllstandsanzeiger
  Anwenderspeicher im BS64 (0000h bis FFFFh).
  DB-RAM im BS65 (0000h bis DD7Fh).
- Systemfehler
  Falls ein Systemfehler auftritt wird
  - der SPS-Prozessor gestoppt
  - alle Ausgänge (P,Q) werden gelöscht
  - im BS7 wird UAPRFE gesetzt (Urlöschen, kein Anlauf)
  - im BS8 wird FE5 gesetzt (Systemfehler)
  - im BS80 erfolgt eine nähere Spezifizierung der Ursache
  - die CPU führt einen Kaltstart durch

Systemfehler-Anzeige

		LEDs		
Fehlerursache	**BS80**	**QVZ**	**ADF**	**ZYK**
Hardware Watchdog	FE01h	O	O	X
CP-SPS Kommunikation	FE02h	O	X	O
Überwachung SPS-Prozessor	FE03h	O	X	X
Tasküberwachung CP-Prozesssor	FE04h	X	O	O
Nicht belegter Interrupt	FE05h	X	O	X
Speicherfehler	FE06h	X	X	O

O → LED aus; X → LED blinkt

Integrierte Sonder-OBs:

OB 110: Zugriff auf das Anzeigenbyte
OB 111: AKKU 1, 2, 3 und 4 löschen
OB 112/113: AKKU-Roll-Up / AKKU-Roll-Down
OB 120: "Alarme gemeinsam sperren" ein-/ausschalten
OB 121: "Weckalarme einzeln sperren" ein-/ausschalten
OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten
OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten
OB 134: *D
OB 135: /D
OB 136: MOD
OB 139: PUSH
OB 150: Systemzeit stellen/lesen
OB 151: Zeit für uhrzeitgesteuerten Weckalarm stellen/lesen
OB 152: Zyklusstatistik
OB 160 bis 163: Zählschleifen
OB 170: Bausteinstack (BSTACK) lesen
OB 180: Variabler Datenbaustein-Zugriff
OB 181: Datenbausteine (DB/DX) testen
OB 182: Datenbereich kopieren
OB 186: Speicher komprimieren
OB 190/OB 192: Merker in DB übertragen
OB 191/OB 193: Datenblöcke in Merkerbereich übertragen
OB 216: Schreiben auf eine Kachel
OB 217: Lesen aus einer Kachel
OB 218: Belegen einer Kachel
OB 220: Vorzeichenerweiterung
OB 221: Zyklusüberwachungszeit einstellen
OB 222: Zyklusüberwachungszeit neu starten
OB 228: Statusinformation einer Programmebene lesen
OB 230: HTB-Send
OB 231: HTB-Receive
OB 232: HTB-Fetch
OB 233: HTB-Control
OB 234: HTB-Reset
OB 235: HTB-Synchron
OB 236: HTB-Send ALL
OB 237: HTB-Receive ALL
OB 250: PID-Algorithmus initialisieren
OB 251: PID-Algorithmus bearbeiten
OB 254/ OB 255: Einen DB verschieben/ duplizieren im DB-RAM

- Die Standardhantierungsbausteine unterstützen die S-Merkerbereiche.
- Datenübertragung mit den Prozeduren 3964 bzw. 3964R und RK512
  Die Datenübertragung erfolgt über die zweite serielle Schnittstelle (SI 2). Die Prozeduren können mit einer höheren Baudrate von bis zu 38400 Baud ausgeführt werden. Die Einstellung erfolgt im DX2 mit dem Wert 000Ah.

Lieferumfang und Gerätevarianten

Grundausführung

- 32-Bit Prozessor für schnelle Bit- und Wortverarbeitung
- 16-Bit Prozessor für Bearbeitung des Betriebsprogramms
- Speicher (RAM) für Merker, S-Merker, Zeiten Zähler, Prozessabbild usw.
- internes DB-RAM 46 kByte bzw. 110 kByte
- interner Speicher (RAM) für Anwenderprogramm 64 kByte bzw. 128 kByte
- PG-Schnittstelle
- zweite serielle frei konfigurierbare Schnittstelle
- Hardware-Uhr

CPU 928B
VIPA 928-3UB12

Grundausführung zusätzlich mit

- Steckplatz für Speicher-EPROM-Module 376

Best.-Nr.:	16 kByte	VIPA 376-0AA11
	32 kByte	VIPA 376-0AA21
	64 kByte	VIPA 376-0AA31

CPU 928B
VIPA 928-3UB21

Grundausführung zusätzlich mit

- Steckplatz für Speicher-Flash-ROM-Module 374

Best.-Nr.:	256 kByte	VIPA 374-2FH21
	512 kByte	VIPA 374-2FJ21

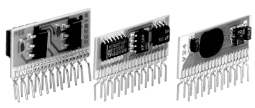
CPU 928C
VIPA 928-3UC11

Grundausführung zusätzlich mit

- internes DB-RAM 110 kByte
- interner Speicher (RAM) für 128 kByte Anwenderprogramm
- Steckplatz für Speicher-Flash-ROM-Modul 374

Best.-Nr.:	256 kByte	VIPA 374-2FH21
	512 kByte	VIPA 374-2FJ21

Hybrid-Module für SI 2



Die physikalische Schnittstelle SI 2 wird standardmäßig als 20mA-Schnittstelle ausgeliefert. Sie kann aber auch über Hybrid-Module beliebig konfiguriert werden. Die physikalischen Schnittstellen können über steckbare Hybrid-Module beliebig konfiguriert werden.

Folgende Hybrid-Module sind bei VIPA erhältlich:

Schnittstelle	Best.-Nr.
RS232C	OP11
RS422	OP13

Teil 2 Hardwarebeschreibung

Überblick

Die CPU 928-Baugruppen erhalten Sie in verschiedenen Ausführungen, auf die in diesem Kapitel mehr eingegangen werden soll. Sie erhalten Informationen zu den Bedienelementen und zur Inbetriebnahme.

Nachfolgend sind beschrieben:

- Komponenten
- Bedien- und Anzeigenelemente
- Anschlüsse
- Inbetriebnahme

Inhalt

Thema	Seite
Teil 2 Hardwarebeschreibung	**2-1**
Aufbau CPU 928	2-2
Bedien- und Anzeigeelemente	2-4
Leuchtdioden	2-7
Schalter	2-9
Serielle Schnittstellen	2-10
Speicher-Module	2-15
Inbetriebnahme	2-16
Anlaufverhalten	2-17
Technische Daten	2-18

Baugruppen-Übersicht

Typ	Bestellnummer	Beschreibung
CPU 928B	VIPA 928-3UB12	CPU 928B (EPROM)
CPU 928B	VIPA 928-3UB21	CPU 928B (Flash-ROM)
CPU 928C	VIPA 928-3UC11	CPU 928C (Flash-ROM)

Aufbau CPU 928

Die CPUs in der Übersicht

Hinweis!

Soweit nichts anderes erwähnt ist, beziehen sich die in diesem Handbuch gemachten Angaben auf die CPU 928-Familie von VIPA!

Komponenten der Baugruppen

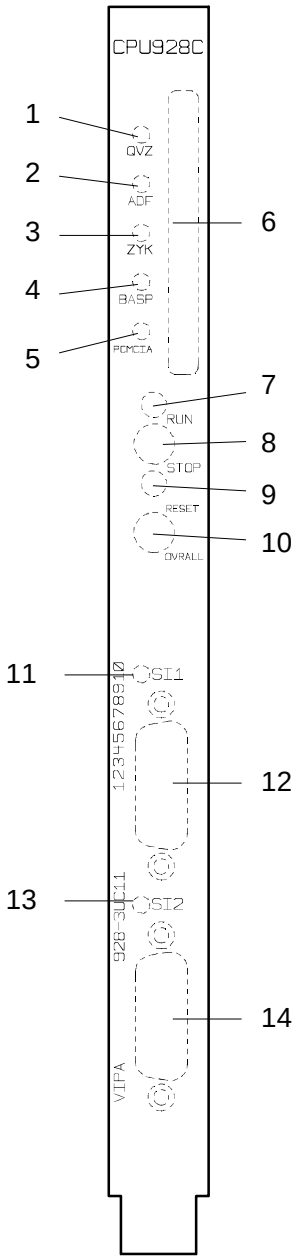
- Recheneinheit, mikroprogrammiert, mit 32 Bit Prozessor für schnelle Bearbeitung von Binär- und Wortanweisungen
- Mikroprozessor (16 Bit) für die Bearbeitung des Betriebsprogramms der Baugruppe und die Bearbeitung der fest eingebauten sowie der seriellen Schnittstellen
- Speicher (RAM) für Merker, S-Merker, Zeiten, Zähler, Prozessabbild usw.
- 928B Interner Speicher (RAM) Batterie gepuffert 46kByte Datenspeicher 64 kByte Programmspeicher + Datenspeicher (Datenbausteine)
- 928C Interner Speicher (RAM) Batterie gepuffert 110kByte Datenspeicher, 128 kByte Programmspeicher
- Hardware-Uhr
- Betriebsprogramm für zyklische, zeit- und prozessgesteuerte Verarbeitung und Fehlerbehandlung
- Sonderfunktionen für Zyklusstatistik

Bedien- und Anzeigeelemente auf der Frontplatte

- Steckplatz für PCMCIA-Speichermodul mit FLASH-ROM (im Programmiergerät zu programmieren) für 64...128kByte Anwenderprogramm. Bei UB 12 EPROM-Schacht bis 64 kByte.
- 15polige SubD-Buchse für PG-Schnittstelle
- 15polige SubD-Buchse als zweite PG-Schnittstelle. Die Schnittstelle kann optional als RS232C-, 20mA-, RS422- oder RS485-Schnittstelle bestückt werden.
- RUN-STOP-Schalter
- RESET/URLÖSCHEN-Taster
- LED (grün) RUN-Zustand
- LED (rot) STOP-Zustand
- LED (rot) QVZ Quittungsverzug/Peripheriefehler
- LED (rot) ADF Adressierfehler
- LED (rot) ZYK Zyklusfehler
- LED (rot) BASP Befehlsausgabesperre
- LED (gelb) PCMCIA Aktivität an PCMCIA
- LED (rot) Fehleranzeige Schnittstelle 1
- LED (rot) Fehleranzeige Schnittstelle 2

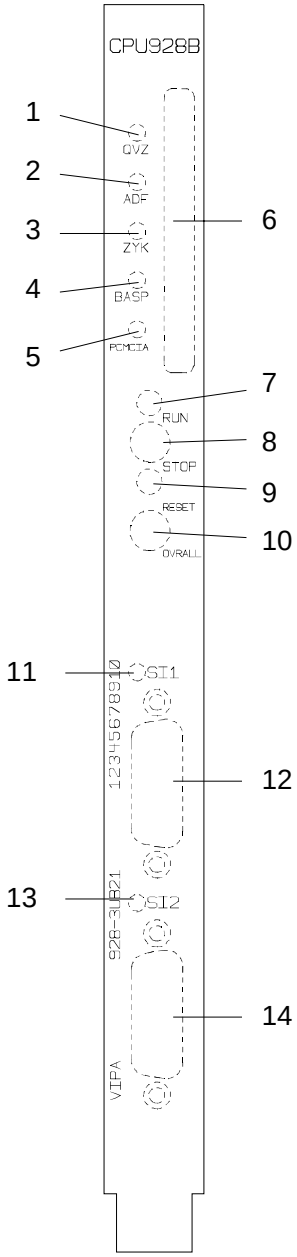
Bedien- und Anzeigeelemente

Frontansicht
CPU 928C
CPU 928-3UC11



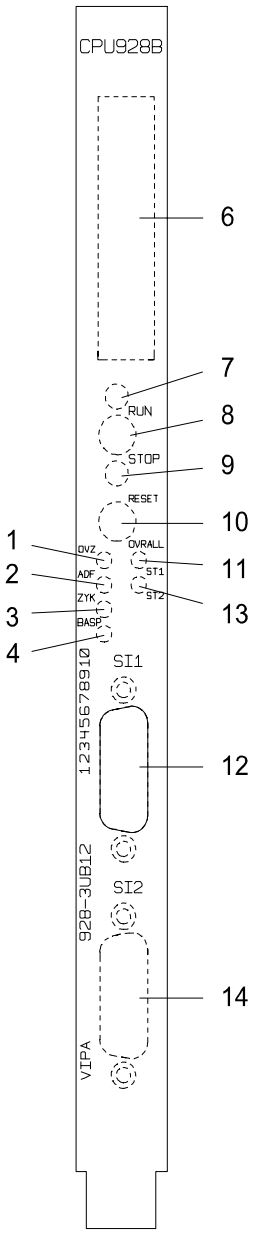
[1] QVZ Quittungsverzug/ Peripheriefehler
[2] ADF Adressierfehler
[3] ZYK Zyklusfehler
[4] BASP Befehlsausgabesperre
[5] PCMCIA Aktivität an PCMCIA
[6] Schacht für Flash-Card 374
[7] RUN-LED
[8] RUN-STOP-Schalter
[9] STOP-LED
[10] RESET/URLÖSCHEN-Taster
[11] Fehler Schnittstelle 1
[12] PG-Schnittstelle SI 1
[13] Fehler Schnittstelle 2
[14] Kommunikations- und PG-Schnittstelle SI 2

Frontansicht
CPU 928B
CPU 928-3UB21



[1] QVZ Quittungsverzug/ Peripheriefehler
[2] ADF Adressierfehler
[3] ZYK Zyklusfehler
[4] BASP Befehlsausgabesperre
[5] PCMCIA Aktivität an PCMCIA
[6] Schacht für Flash-Card 374
[7] RUN-LED
[8] RUN-STOP-Schalter
[9] STOP-LED
[10] RESET/ URLÖSCHEN-Taster
[11] Fehler Schnittstelle 1
[12] PG-Schnittstelle SI 1
[13] Fehler Schnittstelle 2
[14] Kommunikations und PG-Schnittstelle SI 2

Frontansicht
CPU 928B
CPU 928-3UB12



[1] QVZ Quittungsverzug/ Peripheriefehler
[2] ADF Adressierfehler
[3] ZYK Zyklusfehler
[4] BASP Befehlsausgabesperre
[5]
[6] Schacht für Flash-Card 374
[7] RUN-LED
[8] RUN-STOP-Schalter
[9] STOP-LED
[10] RESET/ URLÖSCHEN-Taster
[11] Fehler Schnittstelle 1
[12] PG-Schnittstelle SI 1
[13] Fehler Schnittstelle 2
[14] Kommunikations und PG-Schnittstelle SI 2

Leuchtdioden

QVZ (rot)

LED Quittungsverzug/ Peripheriefehler

- im Anwenderprogramm wird der OB 23 aufgerufen
- bei der Prozessabbildaktualisierung wird der OB 24 aufgerufen

	LED QVZ
an	Bei Direktzugriff oder bei der Prozessabbild-Aktualisierung quittiert eine vom Programm angesprochene Baugruppe nicht mehr, obwohl sie • entweder im Einzelprozessorbetrieb beim Neustart der CPU 928 im Bereich des Prozessabbildes (EB 0 bis 127, AB 0 bis 127) quittiert hat und als vorhanden in der "neunten Spur" eingetragen worden ist. Mögliche Ursachen: • Ausfall der Baugruppe • Ziehen der Baugruppe während des Betriebes, im Stoppzustand oder im ausgeschalteten Zustand ohne anschließenden Neustart.

ADF (rot)

LED Adressierfehler

Das Anwenderprogramm hat eine Peripherieadresse im Prozessabbild angesprochen, unter der keine Baugruppe gesteckt ist.

Bei Zugriff auf das Prozessabbild wird der OB 25 aufgerufen.

ZYK (rot)

LED Zyklusfehler

Die maximale Zyklusüberwachungszeit wurde überschritten.

Bei Überschreitung der Zyklusüberwachungszeit wird der OB 26 aufgerufen.

BASP (rot)

LED Befehlsausgabe sperren

Die Befehlsausgabe ist gesperrt, d.h. ist logisch "0" gesetzt. Z.B. im Betriebszustand STOP. Die digitalen Ausgänge werden direkt in den sicheren Zustand geschaltet.

PCMCIA (gelb)

Die Daten werden von der PCMCIA Card gelesen (nur bei Varianten mit PCMCIA-Modulschacht).

RUN (grün)

RUN-LED
die CPU arbeitet im Normalbetrieb

STOP (rot)

LED STOP zeigt den Betriebszustand STOP der CPU an

LED RUN	LED STOP	Betriebszustand
an	aus	Die CPU ist im Betriebszustand "RUN".
aus	an	Die CPU ist im Betriebszustand "STOP". Nach einer Stoppanforderung per Schalter oder per PG-Funktion zeigt die STOP-LED Dauerlicht, da der Übergang in die Betriebsart "STOP" hier vom Benutzer, angefordert und nicht von der CPU selbst verursacht wurde.
aus	aus	Die CPU ist im Betriebszustand "ANLAUF" oder in Bearbeitungskontrolle.
aus	lang-sames Blinken	Die CPU ist im Betriebszustand "STOP". Die CPU hat einen Übergang in die Betriebsart "STOP" verursacht. Wenn Sie den Betriebsartenschalter auf "STOP" umlegen, geht das Blinken in Dauerlicht über.
aus	schnelles Blinken	Die CPU ist im Betriebszustand "STOP". Urlöschen wurde angefordert. Diese Anforderung kann von der CPU selbst oder per Bedienung erzeugt werden.

SI1/ SI2 (rot)

Fehler Schnittstelle 1/ Schnittstelle 2

LED SI1	LED SI2	Ursache
an	an	An beiden Schnittstellen ist keine Kommunikation möglich. Interner Fehler
an	aus	SI1: Keine Kommunikation möglich. Interner Fehler. SI2: Schnittstelle ist, sofern sie benutzt wird, initialisiert und betriebsbereit.
aus	an	SI1: Schnittstelle ist initialisiert und betriebsbereit SI2: Keine Kommunikation möglich. Falsches Schnittstellenmodul gesteckt oder falsche Para-metrierung der Schnittstelle oder interner Fehler.
aus	aus	Die erste Schnittstelle und, sofern sie benutzt wird, die zweite Schnittstelle sind initialisiert und betriebsbereit.

Schalter

RUN-STOP

RUN-STOP Schalter

Betriebsartenschalter Ein- /Ausschalter der CPU

In Stellung "RUN" bearbeitet die CPU 928 das Anwenderprogramm, wenn die grüne LED "RUN" leuchtet.

Die CPU 928 geht in den Stoppzustand, wenn Sie von "RUN" auf "STOP" schalten. Die rote LED "STOP" leuchtet anschließend.

RESET-OVRALL

Funktionswahlschalter für Urlöschen und Neustart und Wiederanlauf.
Urlöschen

Taster unten halten und RUN-STOP schalten.

Beim Urlöschen werden alle RAM-Bereiche der CPU gelöscht und initialisiert .

Neustart

Taster oben

Beim Neustart werden alle Merker, Zeiten, Zähler und das Prozessabbild gelöscht. Der OB 20 wird aufgerufen. Die Bearbeitung des Anwenderprogramms beginnt von vorne.

Wiederanlauf

Taster in Mittelstellung

Beim Wiederanlauf wird die Bearbeitung des Anwenderprogramms an der Unterbrechungsstelle fortgesetzt. Die Zustände der Merker, Zeiten, Zähler und das Prozessabbild bleiben während des Stillstandes der CPU erhalten.

Hinweis!

Nähere Erläuterungen zu den Betriebszuständen STOP, RUN und ANLAUF finden Sie im Teil "Betriebszustände und Programmbearbeitungsebenen.

Serielle Schnittstellen

Übersicht

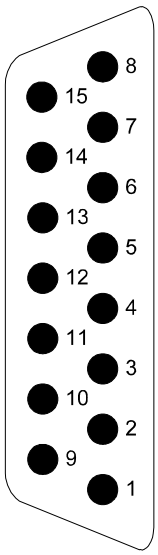
Die CPU 928-Baugruppen besitzen 2 serielle Schnittstellen (SI 1 und SI 2), wobei SI 2 über ein steckbares Hybridmodul von VIPA beliebig konfiguriert werden kann als

- 20mA-Schnittstelle
- RS232C-Schnittstelle
- RS422-Schnittstelle

PG-Schnittstelle SI 1 (fest)

Die Schnittstelle SI 1 dient zur Kommunikation mit Ihrem PG und ist als 20mA-Schnittstelle ausgelegt.

Die Schnittstelle hat folgende Pinbelegung:



Pin	Signal
1	MEXT (externe Masse, Schirm)
2	TTY IN - (Strom-Ausgang)
3	+5V
4	+24V
5	Masse (interne Masse)
6	TTY OUT + (Strom-Eingang)
7	TTY OUT - (Strom-Ausgang)
8	MEXT (externe Masse)
9	TTY IN + (Strom-Eingang)
10	Masse für 24V
11	20mA-Stromquelle des Senders
12	Masse (interne Masse)
13	20mA-Stromquelle des Empfängers
14	+5V
15	Masse (interne Masse)

20mA-Schnittstelle SI 2 (Standard)

Merkmale 20mA auf der CPU 928

- Logisch Zustände als Strompegel
- Datenübertragung je nach Baudrate bis zu 1000m Entfernung
- Datenübertragungsrate bis 38,4kBaud (nur SI 2) bei (3964/RK 512)
- Punkt zu Punkt-Verbindung

Optionen der Betriebsart 20mA

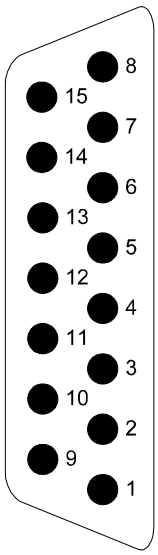
- CPU 928 ist aktiver Teilnehmer und liefert den Linienstrom
- CPU ist passiver Teilnehmer, Gegenstelle übernimmt die Linienstromversorgung.
- SI 2 ist zweiter PG-Kanal (Standard)

Achtung!

Für den aktiven Betrieb stehen auf den S-Signalen Stromquellen zur Verfügung, die entsprechend in die Datenleitungen geschaltet werden müssen.

In diesem Fall ist eine 24V-Spannungsversorgung der Baugruppe erforderlich.

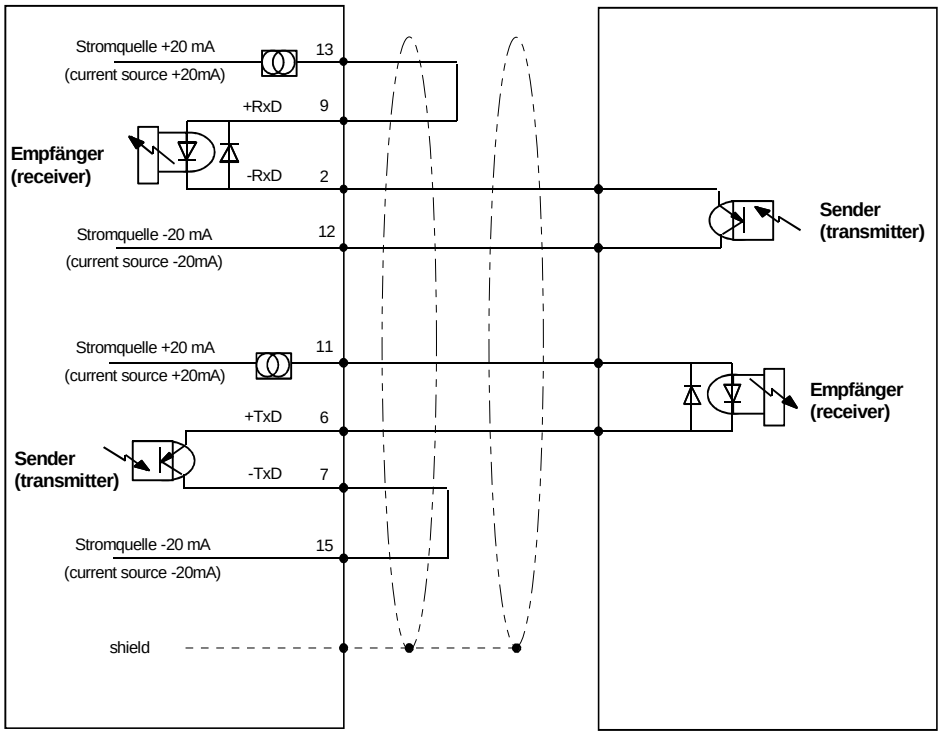
Pinbelegung



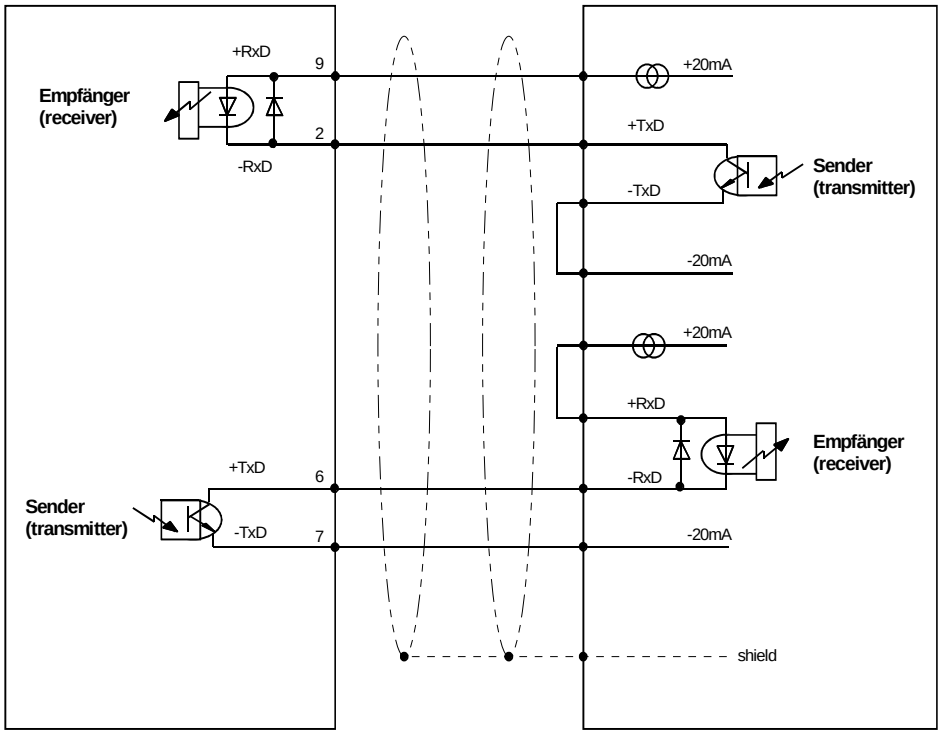
Pin	Signal
1	MEXT (externe Masse, Schirm)
2	TTY IN - (Strom-Ausgang)
3	+5V
4	n. c.
5	n. c.
6	TTY OUT + (Strom-Eingang)
7	TTY OUT - (Strom-Ausgang)
8	n. c.
9	TTY IN + (Strom-Eingang)
10	Masse für 24V
11	20mA-Stromquelle des Senders
12	Masse (interne Masse)
13	20mA-Stromquelle des Empfängers
14	+5V
15	Masse (interne Masse)

Im Gegensatz zur 20mA-Schnittstelle SI 1 für die PG-Kopplung finden Sie hier auf Pin 4 keine 24V. Ansonsten hat die Schnittstelle die gleiche Pinbelegung wie die 20mA Schnittstelle SI 1.

Aktive 20mA-Schnittstelle



Passive 20mA-Schnittstelle

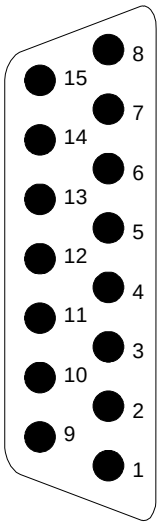


RS232C-Schnittstellen

Merkmale RS232C auf der CPU:

- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplexübertragung in 2-Draht-Technik
- Datenübertragung bis 15m Entfernung
- Datenübertragungsrate bis 38,4kBaud

Pinbelegung

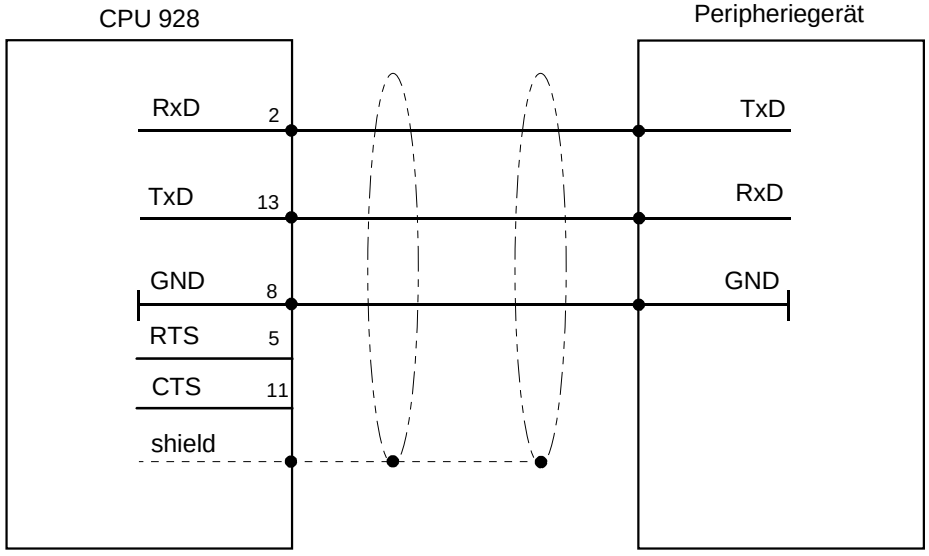


Pin	Signal
1	Schirm
2	RxD
3	n. c.
4	reserviert
5	RTS
6	reserviert
7	reserviert
8	Masse
9	reserviert
10	n. c.
11	CTS
12	n. c.
13	TxD
14	n. c.
15	n.c.

Handshake

Auf der Schnittstelle liegen die Handshake-Signale auf (RTS/CTS). Sie werden aber nicht genutzt.

Kabelbelegung RS232C

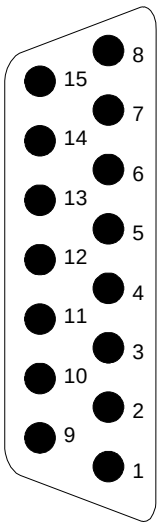


RS422 Schnittstelle

Merkmale RS422 auf der CPU:

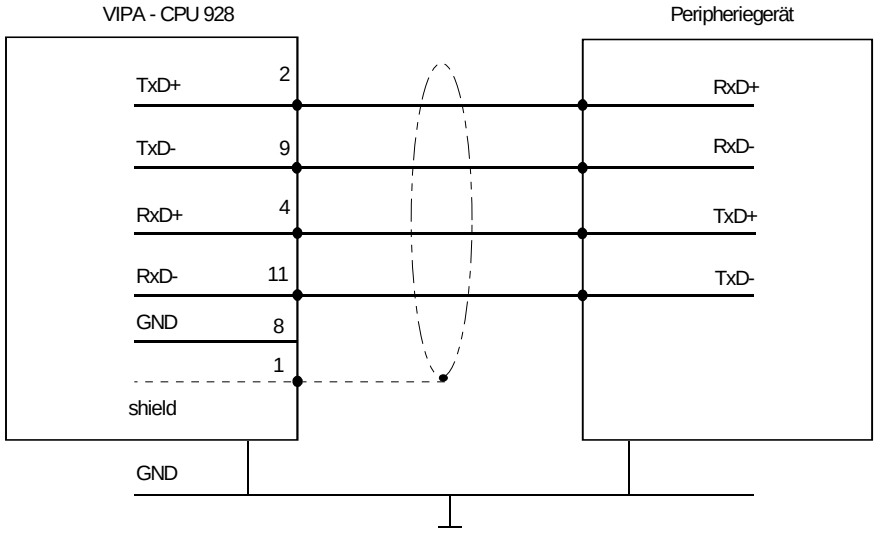
- Logische Zustände als Spannungsdifferenz zwischen zwei verdrillten Adern
- Serielle Punkt-zu-Punkt-Vollduplexübertragung in 4-Draht-Technik
- Multidrop-Verbindung
- Datenübertragung bis 1000m Entfernung
- Datenübertragungsrate bis 115kBaud

Pinbelegung



Pin	Signal
1	Schirm
2	TxD+
3	n. c.
4	RxD+
5	reserviert
6	reserviert
7	reserviert
8	Masse
9	TxD-
10	n. c.
11	RxD-
12	n. c.
13	reserviert
14	n. c.
15	n. c.

Kabelbelegung RS422



Achtung!
Die Massen der Schnittstellen der beiden Geräte müssen verbunden sein.

Speicher-Module

Allgemeines

Code- und Datenbausteine werden durch das Systemprogramm vom Speicher-Modul in den Anwenderspeicher kopiert. Das Systemprogramm setzt außerdem eine Schreibschutzkennung.

Dies bedeutet, dass alle kopierten Bausteine nicht nachgeladen, verändert oder gelöscht werden können.

Um Daten in Datenbausteinen ändern zu können, müssen Sie dafür sorgen, dass die Datenbausteine ins DB-RAM gelangen.

Datenbausteine, die in im Speicher-Modul programmiert sind, können Sie mit Hilfe der OBs 254/255 (nur CPU 928B) in das DB-RAM kopieren.

Sobald die Datenbausteine im DB-RAM liegen, können Sie nachgeladen, geändert oder gelöscht werden.

Änderungen an diesen Datenbausteinen werden jedoch nicht in das Speicher-Modul übernommen. Vor dem nächsten Urlöschen müssen Sie den Inhalt des Speicher-Moduls retten.

Nach dem Urlöschen können Sie das Speicher-Modul ziehen, die Daten stehen im Anwenderspeicher bereit.

Übersicht

Folgende Speicher-Module können in den CPU-928-Baugruppen von VIPA eingesetzt werden:

VIPA-CPU	Modul-Beschreibung	Bestell-Nr.
928B (928-3UB12)	EPROM 16 KB, kurz EPROM 32 KB, kurz EPROM 64 KB, kurz	VIPA 376-0AA11 VIPA 376-0AA21 VIPA 376-0AA31
928B (928-3UB21) 928C (928-UC11)	Flash-ROM 256 KB	VIPA 374-2FH21
928B (928-3UB21) 928C (928-UC11)	Flash-ROM 512 KB	VIPA 374-2FJ21

Hinweis!

Die Memory Card kann nur am PG programmiert werden. Dazu können Sie die PG-Software ab Version 6 verwenden.

Beim Programmieren müssen Sie am PG die Betriebsart "WORD" wählen.

Inbetriebnahme

Achtung!
Schalten Sie die Stromversorgung aus, bevor Sie die Baugruppe ziehen oder stecken.

Stecken

Gehen Sie beim Stecken der CPU in das Zentralgerät folgendermaßen vor:

Schritt	Handlung
1	Entfernen Sie die Abdeckleiste der oberen Verriegelungsschiene indem Sie die 2 Befestigungsschrauben lösen. Drehen Sie den Verriegelungsbolzen an der Front unten waagrecht.
2	Wählen Sie den richtigen Steckplatz aus (VIPA Rack-135U: Steckplatz 2 und 4). Setzen Sie CPUs im AG-135U von Siemens ab Steckplatz 11 ein.
3	Halten Sie die einzubauende Baugruppe an der Frontplatte, Stecken Sie die Baugruppe in die untere und obere Führungsschiene ein und schieben Sie die Baugruppe nach hinten. Die rückwärtigen Stecker der Baugruppe rasten in die Buchse am Rückwandbus ein und der Ausrasthebel an Ihrer Baugruppe wird waagrecht gestellt.
4	Drücken Sie den Verriegelungsbolzen auf der Unterseite der Baugruppe ein und drehen Sie ihn um 90°.
5	Montieren Sie wieder die Abdeckleiste der oberen Verriegelungs-schiene und ziehen Sie die Befestigungsschrauben an.

Ziehen

Gehen Sie beim Ziehen der CPU folgendermaßen vor:

Schritt	Handlung
1	Lösen Sie die obere Verriegelungsschiene des Zentralgerätes.
2	Lösen Sie den Verriegelungsbolzen der Baugruppe.
3	Drücken Sie den Ausrasthebel nach unten und ziehen Sie die Baugruppe dann nach vorne aus dem Zentralgerät heraus.

Anlaufverhalten

Die Baugruppen müssen auf den richtigen Steckplätzen im Zentralgerät stecken. Die Pufferbatterie muss eingebaut und in Ordnung sein, damit die CPU in Betrieb gehen kann.

Erstes Urlöschen

- Stellen Sie den Betriebsartenschalter auf "STOP"
- Schalten Sie die Netzspannung ein.

  → Folgende LEDs müssen auf der CPU leuchten:
  - Rote LED "STOP" (schnell blinkend)
  - Rote LED "BASP".
- Halten Sie den Betriebsartentaster in Stellung "OVERALL RESET" fest, und stellen Sie gleichzeitig den Betriebsartenschalter von "STOP" nach "RUN".

  → Die rote LED "STOP" zeigt nun Dauerlicht.

Neustart

- Stellen Sie den Betriebsartenschalter auf "STOP".
- Halten Sie den Betriebsartentaster in Stellung "RESET" fest, und stellen Sie gleichzeitig den Betriebsartenschalter von "STOP" nach "RUN".

  → Rote LED "STOP" erlischt,
  Grüne LED "RUN" leuchtet
  Rote LED "BASP" erlischt.

Die CPU ist jetzt im Betriebszustand "RUN", aber noch ohne Anwenderprogramm.

Wiederanlauf

Mit dem Betriebsartenschalter können Sie weiterhin einen manuellen Wiederanlauf der CPU 928 ausführen.

Stellen Sie den Betriebsartenschalter von "STOP" nach "RUN".

→ Rote LED "STOP" erlischt
Grüne LED "RUN" leuchtet
Rote LED "BASP" erlischt.

Für Wartungszwecke oder im Fehlerfall kann mit der beschriebenen Inbetriebnahme ohne Anwenderprogramm festgestellt werden, ob die CPU fehlerfrei arbeitet.

Technische Daten

Mikroprozessoren	68EC040 (32bit)
	68340 (16bit)
Speicherplatz	
RAM intern (nur DBs)	46 kByte , 110kByte (928C)
RAM intern (alle Bausteine)	64 kByte , 128kByte (928C)
Speichermedium	EPROM , max. 64kByte (928B-3UB12)
	Flash-ROM max. 64kByte (928B-3UB21)
Bearbeitungszeit für	
1 K Binäranweisung	1 ms
1 K Wortanweisung	1 ms
Zykluszeitüberwachung	per Programm einstellbar, Voreinstellung: 200ms
Merker	2048
S-Merker	8192
Zeitglieder	
Anzahl	256
Zeitbereich	0.01 ... 9990s
Zähler	
Anzahl	256
Zählbereich	0 ... 999
Eingänge	
digital	max 1024 mit Prozessabbild
zusätzlich	max. 3072 ohne Prozessabbild
analog	max. 192
Ausgänge	
digital	max 1024 mit Prozessabbild
zusätzlich	max. 3072 ohne Prozessabbild
analog	max. 192
Programmbearbeitung	
	Zyklus (OB1, FB0)
	alarmgesteuert
	zeitgesteuert in 9 Takten
	uhrzeitgesteuert
Stromaufnahme (bei 5V)	
CPU 928B	typ. 1,8 A
Speichermodul	typ. 0,1 A
Verlustleistung	max 10 W
Platzbedarf	1 Steckplatz

Teil 3 Anwenderprogramm

Überblick

Im nachfolgenden Kapitel erfahren Sie, aus welchen Komponenten sich ein Anwenderprogramm für die CPU 928 zusammensetzt und wie es strukturiert werden kann.

Nachfolgend sind beschrieben:

- Strukturierte Programmierung
- Operationen
- Zahlendarstellung
- Organisationsbausteine für Anwenderschnittstellen
- Sonderfunktions-Organisationsbausteine
- Aufbau und Programmierung von Funktionsbausteinen
- Erstellung und Aufschlagen von Datenbausteinen
- Spezielle Datenbausteine

Inhalt

Thema **Seite**

Teil 3 Anwenderprogramm 3-1
Programmiersprache 3-2
Programm-, Organisations- und Schrittbausteine 3-13
Funktionsbausteine 3-19
Datenbausteine 3-29

Programmiersprache

Mit der Programmiersprache STEP® 5 von Siemens setzen Sie die Ihnen vorliegenden Automatisierungsaufgaben in Programme um, die in Ihren Automatisierungsgeräten ablaufen.

In dieser Programmiersprache können Sie sowohl einfache binäre Funktionen als auch komplexe digitale Funktionen und arithmetische Operationen einschließlich Gleitpunktarithmetik programmieren.

Der Befehlsumfang der Programmiersprache gliedert sich in

- Grundoperationen:
  - in allen Code-Bausteinen anwendbar,
  - Darstellungsarten Kontaktplan (KOP), Funktionsplan (FUP), Anweisungsliste (AWL).
- Ergänzende Operationen und Systemoperationen:
  - nur in Funktionsbausteinen anwendbar,
  - Darstellungsart nur Anweisungsliste (AWL),
  - Systemoperationen: nur für Anwender mit sehr guten Systemkenntnissen.

Darstellungsarten KOP, FUP, AWL

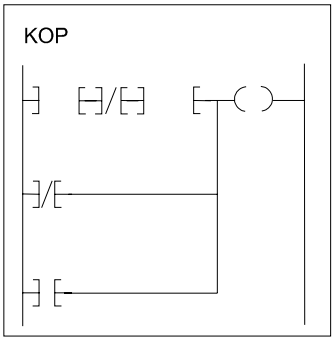
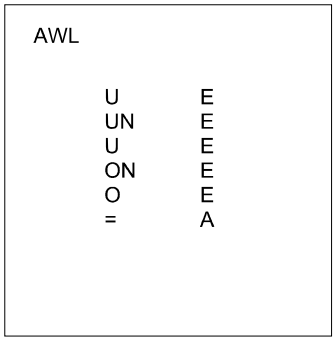
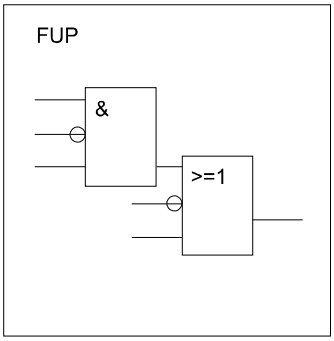
Beim Programmieren können Sie für jeden einzelnen Code-Baustein zwischen den drei Darstellungsarten Kontaktplan (KOP), Funktionsplan (FUP) und Anweisungsliste (AWL) wählen, so dass die Programmiermethode dem jeweiligen Anwendungsfall angepasst werden kann.

Der von den Programmiergeräten (PG) erzeugte Maschinencode MC 5 ist bei den drei Darstellungsarten identisch.

Wenn Sie beim Programmieren bestimmte Regeln berücksichtigen, kann das PG Ihr Anwenderprogramm von einer Darstellungsart in jede andere übersetzen!

Graphische Darstellungen oder Liste mit Anweisungen

Während Sie mit den Darstellungsarten FUP und KOP die Möglichkeit haben, Ihr Programm graphisch darzustellen, werden in der Anweisungsliste die einzelnen Befehle aufgelistet.

Kontaktplan	Anweisungsliste	Funktionsplan
Programmieren mit grafischen Symbolen wie Stromlaufplan entspricht DIN 19239	Programmieren mit mnemotechnischen Abkürzungen der Funktionsbezeichnungen entspricht DIN 19239	Programmieren mit graphischen Symbolen entspricht IEC 117-15 DIN 40700 DIN 40719 DIN 19239
		

Grafische Darstellung von Ablaufsteuerungen

GRAPH 5/6 von Siemens ist eine Programmiersprache zur graphischen Darstellung von Ablaufsteuerungen. Sie ist den Darstellungsarten KOP, FUP und AWL übergeordnet. Dieses geschriebene Programm in grafischer Darstellung wird vom PG automatisch in ein SPS-Programm umgesetzt.

Strukturierte Programmierung

Mit STEP® 5 von Siemens kann das Anwenderprogramm strukturiert werden. Es wird dabei in einzelne, in sich geschlossene Programmabschnitte (Bausteine) aufgeteilt.

Die Gliederung Ihres Anwenderprogramms verdeutlicht somit auf den ersten Blick die wesentlichen Programmstrukturen oder hebt programmtechnisch zusammenhängende Anlagenteile hervor.

Vorteile

Dieses Verfahren des "strukturierten Programmierens" bietet Ihnen folgende Vorteile:

- einfache und übersichtliche Programmierung auch umfangreicher Programme
- Möglichkeit zum Standardisieren von Programmteilen
- einfache Programmorganisation
- leichte Änderungsmöglichkeiten
- einfacher, abschnittsweiser Programmtest
- einfache Inbetriebnahme

Was ist ein Baustein?

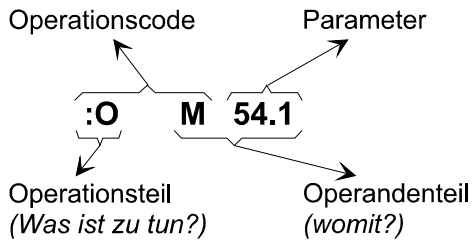
Ein Baustein ist ein durch Funktion, Struktur oder Verwendungszweck abgegrenzter Teil des Anwenderprogramms.

Man unterscheidet Bausteine, die Anweisungen (Code) enthalten (Organisationsbausteine, Programmbausteine, Funktionsbausteine, Schrittbausteine) und Bausteine, die Daten enthalten (Datenbausteine).

Operationen

Eine Operation ist die kleinste selbständige Einheit des Anwenderprogramms. Sie ist die Arbeitsvorschrift für die CPU. Eine Operation setzt sich zusammen aus einem Operationsteil und einem Operandenteil.

Beispiel



Absolute und symbolische Operanden

Den Operandenteil können Sie entweder absolut oder symbolisch (über die Zuordnungsliste) eingeben.

Beispiel für die absolute Darstellung: :U E1.4

Beispiel für die symbolische Darstellung: :U -Motor1

Beachten Sie zur absoluten und symbolischen Programmierung die Bedienungsanleitung zu Ihrem PG.

Anwendung der Operationen

Der Operationsumfang ermöglicht Ihnen

- binäre Werte miteinander zu verknüpfen, zu setzen oder rückzusetzen,
- Werte zu laden und zu transferieren,
- Werte miteinander zu vergleichen und arithmetisch zu bearbeiten,
- Zeit- und Zählwerte vorzugeben,
- Zahlendarstellungen umzuwandeln,
- Bausteine aufzurufen und Sprünge innerhalb eines Bausteins auszuführen
- die Programmbearbeitung zu beeinflussen.

Verknüpfungsergebnis VKE

Das zentrale Bit für die Programmsteuerung ist das Verknüpfungsergebnis VKE. Es wird durch binäre Verknüpfungen gebildet sowie von einigen Operationen beeinflusst.

Der gesamte Operationsvorrat sowie Einzelheiten zur VKE-Bildung werden ausführlich in Teil "Programmbearbeitung" beschrieben. Sie finden dort Programmierbeispiele zu den einzelnen Befehlen.

Zahlendar-stellung

Damit die CPU Zahlenwerte miteinander verknüpfen, verändern oder vergleichen kann, müssen diese in einer binärcodierten Darstellung in die Akkumulatoren (Arbeitsregister der CPU) geladen werden.

Abhängig von der auszuführenden Operation sind folgende Zahlendarstellungen zulässig:

- **Dualzahlen:** 16-Bit-Festpunktzahlen,
  32-Bit-Festpunktzahlen,
  Gleitpunktzahlen mit 16- oder 32-Bit-Mantisse.
- **Dezimalzahlen:** BCD-codierte Zahlen (Vorzeichen und drei Dezimalstellen)

Zahleneingabe am PG

Bei der Ein- und Ausgabe von Zahlenwerten stellen Sie am Programmiergerät das Datenformat (z.B. KF für Festpunkt) ein, in dem Sie den Zahlenwert eingeben bzw. angezeigt haben möchten. Auf diese Weise übernimmt das PG die Umrechnung aus der intern verwendeten Zahlendarstellung in die von Ihnen gewünschte Darstellungsart.

Zulässige Operatoren

Mit den 16-Bit-Festpunktzahlen und Gleitpunktzahlen können Sie alle arithmetischen Operationen wie Vergleichen, Addieren, Subtrahieren, Multiplizieren und Dividieren ausführen.

Mit den 32-Bit-Festpunktzahlen werden Vergleichsoperationen durchgeführt. Außerdem werden sie bei der Umwandlung von BCD-codierten Zahlen in Gleitpunktzahlen als Zwischenstufe benötigt. Mit den Befehlen +D und -D können sie auch für Additionen und Subtraktionen verwendet werden.

Die Sprache enthält Umwandlungsoperationen, mit denen Sie Zahlen direkt in die wichtigsten Zahlendarstellungen umwandeln können.

Hinweis!

BCD-codierte Zahlen dürfen nicht für arithmetische Operationen verwendet werden, da sie zu falschen Ergebnissen führen

16- und 32-Bit-Festpunktzahlen

Festpunktzahlen sind ganze, mit einem Vorzeichen versehene Dualzahlen.

Codierung der Festpunktzahlen

Sie sind in binärcodierter Darstellung 16 Bit (=1 Wort) bzw. 32 Bit (=2 Wörter) breit, wobei Bit-Nr.15 bzw. Bit-Nr.31 das Vorzeichen enthält:

- '0'= positive Zahl,
- '1'= negative Zahl.

Negative Zahlen werden im 2er-Komplement dargestellt.

PG-Eingabe

16-Bit-Festpunktzahlen:	KF
32-Bit-Festpunktzahlen:	KF

Zulässiger Zahlenbereich

16-Bit-Festpunktzahl:	-32768 bis +32767
32-Bit-Festpunktzahl:	-2147483648 bis +2147483647

Anwendung der Festpunktzahlen

Festpunktzahlen werden bei einfachen Rechenaufgaben und beim Vergleich von Zahlenwerten verwendet. Da Festpunktzahlen immer ganze Zahlen sind, beachten Sie bitte, dass das Ergebnis einer Division von zwei Festpunktzahlen ebenfalls eine Festpunktzahl ohne Dezimalstellen ist.

Gleitpunktzahlen

Gleitpunktzahlen sind positive und negative gebrochene Zahlen. Sie belegen immer ein Doppelwort (32 Bit). Eine Gleitpunktzahl wird als Exponentialzahl dargestellt.

Die CPU 928 rechnet beim Addieren, Subtrahieren, Multiplizieren und Dividieren mit einer 24bit breiten Mantisse (Bit 0 bis 23).

Der Exponent ist 8 Bit lang. Der Exponent gibt die Größenordnung der Gleitpunktzahl an. Am Vorzeichen des Exponenten erkennen Sie, ob der Betrag der Gleitpunktzahl größer oder kleiner als 0,1 ist.

Anwendung der Gleitpunktzahlen

Verwenden Sie Gleitpunktzahlen für die Lösung umfangreicherer Rechenaufgaben, insbesondere bei Multiplikationen und Divisionen, und dann, wenn Sie mit sehr großen oder sehr kleinen Zahlen arbeiten.

Genauigkeit

Die Mantisse gibt die Genauigkeit der Gleitpunktzahl an:

Genauigkeit bei einer 24-Bit-Mantisse:

2^{-24} = 0,000000059604 (entspricht 7 Nachkommastellen)

Ist das Vorzeichen der Mantisse '0', ist die Zahl positiv; bei Vorzeichen '1' ist es eine negative Zahl in 2er-Komplement-Darstellung.

Der Gleitpunktwert '0' wird als dualer Wert 80000000H (32 Bit, siehe "Codierung der Gleitpunktzahlen") dargestellt.

Codierung der Gleitpunktzahlen

Eine Gleitpunktzahl wird folgendermaßen codiert:

31	30 24	23	22 0
V	2^6 ... ...2^0	V	2^{-1} ... ...2^{-23}
Exponent		Mantisse	

Angabe des Datenformats für Gleitpunktzahlen am PG:KG

Zulässiger Zahlenbereich

$\pm 0{,}1469368 \times 10^{-38}$ bis $\pm 0{,}1701412 \times 10^{39}$

Ein-/Ausgabe am PG

a) im Codebaustein:

Es soll die Zahl Z = 12,34567 als Gleitpunktkonstante geladen werden.

Eingabe:

:LKG1234567+2

Ausgabe durch PG nach Übernahme der Zeile:

:L KG+ 1234567 + 02

Mantisse mit Vorzeichen — Exponent (Basis 10) mit Vorzeichen

Wert der eingegebenen Zahl: $+0{,}1234567 \times 10^{+2} = 12{,}34567$

b) im Datenbaustein:

Es soll die Zahl Z = 0,005 als Gleitpunktkonstante definiert werden.

Eingabe:

6: KG = 5000000 - 02

Ausgabe durch PG nach Übernahme der Zeile:

6: KG = 5000000 - 02

Mantisse mit Vorzeichen — Exponent (Basis 10) mit Vorzeichen

Wert der eingegebenen Zahl: $0{,}5 \times 10^{-2} = 0{,}005$

BCD (Binär-codierte Dezimalzahl)

Dezimalzahlen werden als BCD-Zahlen dargestellt. Mit Vorzeichen und 3 Ziffern belegen sie im Akkumulator 16 Bit (1 Wort):

15 12	11 8	7 4	3 0
V V V V	Hunderter	Zehner	Einer

Die einzelnen Ziffern sind positive 4-Bit-Dualzahlen zwischen 0000 und 1001 (0 und 9 dezimal).

Die linken Bit sind für das Vorzeichen reserviert.

Vorzeichen bei einer positiven Zahl: 0000

Vorzeichen bei einer negativen Zahl: 1111

Zulässiger Zahlenbereich

-999 bis +999

Bausteine und deren Ablage im Speicher

Kennzeichnung

Ein Baustein ist gekennzeichnet durch:

- den Bausteintyp (OB,PB,SB,FB,FX,DB,DX)
- die Bausteinnummer (Zahl zwischen 0 und 255).

Baustein-Typen

Die Programmiersprache unterscheidet folgende Bausteintypen:

Organisationsbausteine (OB):

Die Organisationsbausteine sind die Schnittstelle zwischen dem Systemprogramm und dem Anwenderprogramm. Sie können in zwei Gruppen unterteilt werden:

Mit den OB 1 bis 39 können Sie die Programmbearbeitung, das Anlaufverhalten der CPU und das Verhalten im Fehlerfall steuern, indem Sie diese Bausteine entsprechend programmieren. Diese OB werden vom Systemprogramm aufgerufen.

Die OB 40 bis 255 enthalten Sonderfunktionen des Systemprogramms. Sie werden bei Bedarf vom Anwenderprogramm aufgerufen.

Programmbausteine (PB):

Programmbausteine werden zur Strukturierung des Anwenderprogramms verwendet und enthalten die nach technologischen oder funktionellen Gesichtspunkten gegliederten Teilprogramme. Die PB bilden den Kern des Anwenderprogramms.

Schrittbausteine (SB):

Schrittbausteine waren ursprünglich spezielle Programmbausteine zur schrittweisen Bearbeitung von Ablaufketten. Ablaufketten können jedoch inzwischen über GRAPH 5/6 (von Siemens) programmiert werden. Daher haben Schrittbausteine nicht mehr ihre ursprüngliche Bedeutung. Schrittbausteine sind jetzt eine zahlenmäßige Erweiterung der Programmbausteine. Sie können wie diese eingesetzt werden.

Funktionsbausteine(FB/FX):

Funktionsbausteine dienen zum Programmieren von häufig wiederkehrenden oder auch von komplexen Funktionen (z.B. digitale Funktionen, Ablaufsteuerungen, Regelungen, Meldefunktionen).Ein Funktionsbaustein kann von übergeordneten Bausteinen mehrfach aufgerufen werden und bei jedem Aufruf mit neuen Operanden versorgt ("parametriert") werden. Durch die Verwendung der Bausteinart FX wird die Zahl der maximal möglichen Funktionsbausteine von 256 auf 512 erhöht.

Datenbausteine (DB/DX):

In Datenbausteinen stehen die (festen oder veränderbaren) Daten, mit denen das Anwenderprogramm arbeitet. Diese Bausteinart enthält keine Anweisungen und unterscheidet sich in ihrer Funktion grundsätzlich von den übrigen Bausteinen. Durch die Verwendung der Bausteinart DX wird die Zahl der maximal möglichen Datenbausteine verdoppelt.

Aufbau der Bausteine

Alle Bausteintypen bestehen aus

- Bausteinkopf
- Bausteinrumpf.

Bausteinkopf

Der Bausteinkopf hat immer eine Länge von 5 Wörtern und enthält Informationen für die Bausteinverwaltung im PG und Daten für das Systemprogramm.

Bausteinrumpf

Im Bausteinrumpf sind -abhängig vom Bausteintyp- enthalten:

- Befehle (bei OB, PB, SB, FB, FX),
- variable oder konstante Daten (bei DB, DX)
- Formaloperandenliste (bei FB, FX).

Bausteinvorkopf

Zu den Bausteinen vom Typ DB, DX, FB und FX erzeugt das Programmiergerät zusätzlich einen Bausteinvorkopf (DV, DXV, FV, FXV). Diese Bausteinvorköpfe enthalten Informationen über das Datenformat (bei DB und DX) bzw. über die Sprungmarken (bei FB und FX), die nur das Programmiergerät auswerten kann. Die Bausteinvorköpfe werden deshalb nicht in den CPU-Speicher übertragen.

Als Anwender haben Sie auf den Inhalt eines Bausteinvorkopfes direkt keinen Einfluss.

Maximale Länge

Ein Baustein darf maximal 4096 Wörter (1 Wort entspricht 16 Bit) im Programmspeicher der CPU belegen.

Verfügbare Bausteine

Von den einzelnen Bausteintypen stehen Ihnen zum Programmieren zur Verfügung:

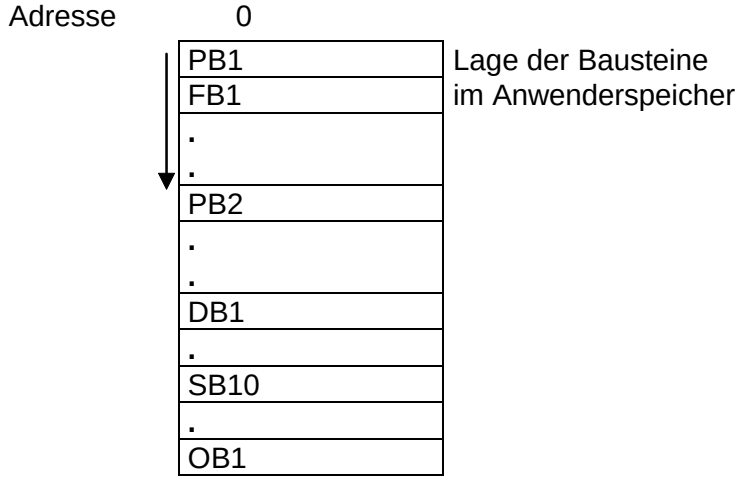
OB	1	bis	39	
FB	0	bis	255	insgesamt 512
FX	0	bis	255	
PB	0	bis	255	
SB	0	bis	255	
DB	3	bis	255	insgesamt 506
DX	3	bis	255	

Die Datenbausteine DB 0, DB 1, DB 2, DX 0, DX 1 und DX 2 enthalten Parameter. Sie sind für bestimmte Funktionen reserviert und deshalb nicht beliebig verwendbar.

Ablage der Bausteine

Alle programmierbaren Bausteine werden vom PG in der Reihenfolge ihres Transfers im Programmspeicher hinterlegt. Mit der PG-Funktion "Übertragen Bausteine B" werden zunächst die Code- und dann die Datenbausteine zum AG übertragen. Bei RAM-Betrieb wird zunächst der Anwenderspeicher nach Übertragung der Codebausteine mit Datenbausteinen aufgefüllt, und danach werden die restlichen Datenbausteine in das DB-RAM geschrieben. Bei der 928C werden Datenbausteine nur im DB-RAM hinterlegt.

Die Anfangsadressen der gespeicherten Bausteine werden im Datenbaustein DB 0 hinterlegt.



Alternatives Laden (nur bei Version 3UB12)

Durch Setzen von Bit 0 des Systemdatums BS 144 können Sie Datenbausteine vorrangig (d.h. solange Platz vorhanden ist) in das DB-RAM laden ("alternatives Laden" /BS 144). Erst wenn das DB-RAM gefüllt ist, werden Datenbausteine in den Anwenderspeicher übertragen.

Korrigieren und Löschen von Bausteinen

Beim Korrigieren von Bausteinen im "RAM-Betrieb" wird der "alte" Baustein im Speicher für ungültig erklärt und ein neuer Baustein im Speicher eingetragen.

Ebenso werden beim Löschen von Bausteinen die Bausteine nicht wirklich gelöscht, sondern nur für ungültig erklärt. Gelöschte und korrigierte Bausteine belegen also weiterhin Speicherplatz.

Hinweis!

Mit der Online-Funktion SPEICHER KOMPRIMIEREN schaffen Sie Speicherplatz für neue Bausteine: Die Funktion optimiert die Speicherplatzbelegung, indem sie ungültig markierte Bausteine löscht und die gültigen Bausteine zusammenschiebt. Das Zusammenschieben erfolgt getrennt nach Anwenderspeicher und DB-RAM.

Programm-, Organisations- und Schrittbausteine

Programmbausteine (PB), Organisationsbausteine (OB) und Schrittbausteine (SB) unterscheiden sich hinsichtlich Programmierung und Aufruf nicht. Alle drei können Sie wahlweise in den Darstellungsarten KOP, FUP und AWL programmieren.

Programmieren

Gehen Sie beim Programmieren von Programm-, Organisations- und Schrittbausteinen folgendermaßen vor:

1. Geben Sie zuerst den Bausteintyp, dann die Nummer des Bausteins an, den Sie programmieren wollen.

   folgende Nummern stehen Ihnen jeweils zur Verfügung:

Programmbausteine	0	bis	255
Organisationsbausteine	1	bis	39
Schrittbausteine	0	bis	255

2. Geben Sie Ihr Anwenderprogramm in ein.

   Beim Programmieren von PB, OB und SB dürfen Sie nur die Grundoperationen verwenden!

   Ein Baustein sollte immer einen in sich abgeschlossenen Programmabschnitt enthalten. Logische Verknüpfungen müssen innerhalb eines Bausteins abgeschlossen sein.

3. Schließen Sie die Programmeingabe mit der Anweisung "BE" (Bausteinende) ab.

Aufrufen

Bausteine (-außer OB 1 bis OB 39-) müssen zum Bearbeiten aufgerufen werden. Dies geschieht durch spezielle Operationen, die Bausteinaufrufe.

Diese Bausteinaufrufe können innerhalb eines Organisations-, Programm-, Funktions- oder Schrittbausteins programmiert werden. Sie sind vergleichbar mit Sprüngen in ein Unterprogramm. Jeder Sprung verursacht einen Bausteinwechsel. Die Rücksprungadresse im aufrufenden Baustein wird vom System zwischengespeichert.

Aufrufe können sowohl unbedingt als auch bedingt ausgeführt werden:

Unbedingter Aufruf

Die Anweisung SPA gehört zu den unbedingten Operationen. Sie hat selbst keinen Einfluss auf das VKE. Dieses wird beim Sprung in den neuen Baustein mitgenommen. Dort kann es zwar ausgewertet, jedoch nicht mehr weiter verknüpft werden.

Der angesprochene Baustein wird unabhängig vom Verknüpfungsergebnis bearbeitet.

Beispiel: **SPA** PB 100

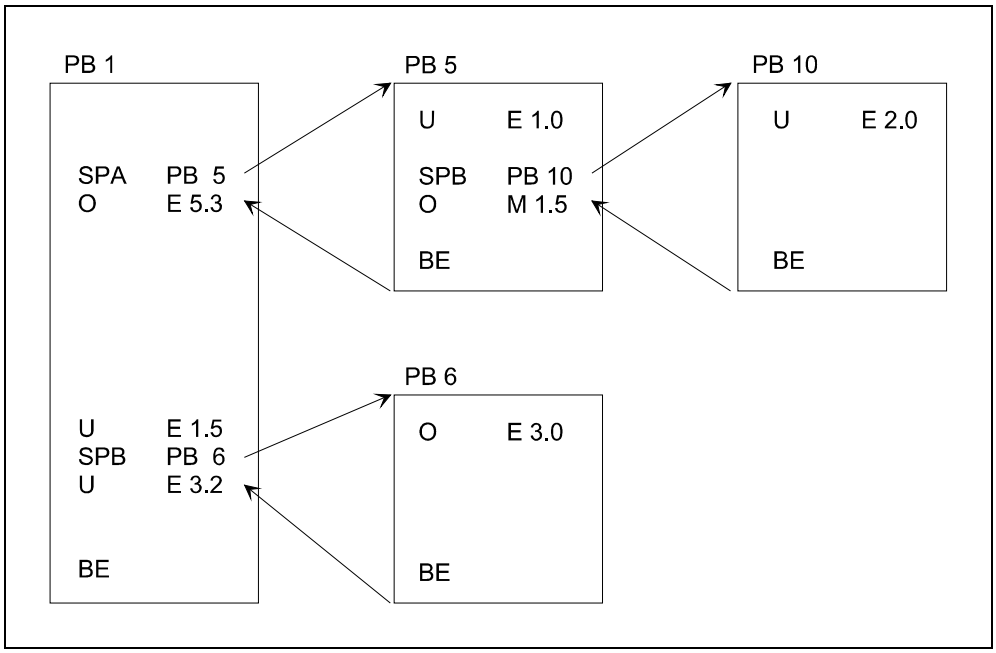
Bedingter Aufruf

Die Anweisung SPB gehört zu den bedingten Operationen, d.h., der angesprochene Baustein wird nur bearbeitet, wenn das Verknüpfungsergebnis **VKE** = **1** ist. Bei VKE = 0 wird die Sprunganweisung nicht ausgeführt.

Beispiel: **SPB** PB 100

Hinweis!

Nach Ausführen der bedingten Sprungoperation ist das VKE auf '1' gesetzt, unabhängig davon, ob der Sprung in den aufgerufenen Baustein ausgeführt wird oder nicht.



Was die BE-Anweisung bewirkt

Nach der Anweisung BE (Bausteinende) setzt die CPU das Anwenderprogramm in dem Baustein fort, in dem der Bausteinaufruf programmiert wurde: Die Programmbearbeitung wird bei der auf den Bausteinaufruf folgenden Anweisung fortgesetzt.

Die Anweisung BE wird unabhängig vom Verknüpfungsergebnis bearbeitet. Nach BE kann das Verknüpfungsergebnis nicht mehr weiter verknüpft werden. Das bei Ausführung des BE-Befehls vorhandene Verknüpfungs-/Rechenergebnis wird jedoch an den aufrufenden Baustein übergeben und kann dort ausgewertet werden. Bei der Rückkehr aus dem aufgerufenen Baustein werden die Inhalte von AKKU 1, AKKU 2, AKKU 3 und AKKU 4, die Anzeigen ANZ 0 und ANZ 1 und das Verknüpfungsergebnis VKE nicht verändert (nähere Erläuterungen zu den AKKUs, ANZ 0/ANZ 1 und VKE siehe Abschnitt "Programmbearbeitung").

Organisationsbausteine für Anwenderschnittstellen

Schnittstelle zwischen dem Systemprogramm und dem Anwenderprogramm sind die Organisationsbausteine. Die Organisationsbausteine OB 1 bis OB 39 sind Teile des Anwenderprogramms, die Sie genauso wie Programmbausteine programmieren. Durch Programmierung dieser OBs können Sie das Verhalten der CPU beim Anlauf, während der Programmbearbeitung und im Fehlerfall beeinflussen. Die Organisationsbausteine sind wirksam, sobald sie in den AG-Speicher geladen werden. Dies ist auch im laufenden Betrieb möglich.

Nachdem das Systemprogramm den betreffenden Organisationsbaustein aufgerufen hat, wird das darin enthaltene Anwenderprogramm bearbeitet.

Hinweis!

Die Organisationsbausteine OB 1 bis OB 39 für Anwenderschnittstellen werden vom Anwender programmiert und vom Systemprogramm als Reaktion auf bestimmte Ereignisse automatisch aufgerufen.

Zu Testzwecken können diese Organisationsbausteine auch vom Anwenderprogramm aufgerufen werden (SPA/SPB OB xxx). Es ist jedoch nicht möglich, z.B. durch Aufruf von OB 20 einen NEUSTART auszulösen

Die nachfolgenden Tabellen geben Ihnen eine Übersicht über die Anwenderschnittstellen (OBs).

OBs zur Steuerung der Programmbearbeitung

Baustein	Funktion und Aufrufkriterium
OB 1	Organisation der zyklischen Programmbearbeitung: erster Aufruf nach Ende einer Anlaufart, dann zyklischer Aufruf
OB 2	Organisation der alarmgesteuerten Programmbearbeitung; Aufruf durch Interrupt-Signal des Rückwandbus (Prozessalarm)
OB 3 bis 5	bei CPU 928 nicht vorhanden
OB 7, OB 8	bei CPU 928 nicht vorhanden
OB 9	Uhrzeitgesteuerter Weckalarm
 OB 10 OB 11 OB 12 OB 13 OB 14 OB 15 OB 16 OB 17 OB 18	Weckalarm mit festen Zeitrastern Aufruf alle 10 ms Aufruf alle 20 ms Aufruf alle 50 ms Aufruf alle 100 ms Aufruf alle 200 ms Aufruf alle 500 ms Aufruf alle 1 s Aufruf alle 2 s Aufruf alle 5 s

OBs zur Steuerung des Anlaufverhaltens

Baustein	Funktion und Aufrufkriterium
OB 20	Aufruf bei NEUSTART (manuell und automatisch)
OB 21	Aufruf bei MANUELLEM WIEDERANLAUF
OB 22	Aufruf bei AUTOMATISCHEM WIEDERANLAUF

OBs für Reaktionen auf Geräte oder Programmfehler

Ist im Fehlerfall der OB nicht geladen, so geht die CPU in den Stoppzustand.

AUSNAHME: Bei nicht vorhandenem OB 23, OB 24 wird QVZ-LED kurz gesetzt.

Baustein	Funktion und Aufrufkriterium
OB 19	Laufzeitfehler (LZF): Aufruf eines nicht geladenen Bausteins
OB 23	Quittungsverzug (QVZ) im Anwenderprogramm (bei Direktzugriff auf Peripheriebaugrupppen oder andere Rückwand-Bus-Adressen)
OB 24	Quittungsverzug (QVZ) beim Aktualisieren des Prozessabbildes
OB 25	Adressierfehler (ADF)
OB 26	Zykluszeitüberschreitung (ZYK)
OB 27	Befehlscodefehler (BCF): Operationscode nicht zulässig
OB 28	STOP durch PG-Funktion/Stoppschalter/Rückwand-Bus [1)]
OB 29	Befehlscodefehler (BCF): Operationscode nicht zulässig
OB 30	Befehlscodefehler (BCF): Parameter nicht zulässig
OB 31	sonstige Laufzeitfehler (LZF)
OB 32	Laufzeitfehler (LZF): Lade- und Transferfehler bei Datenbausteinen
OB 33	Weckfehler (WECK-FE)
OB 34	Fehler bei Reglerbearbeitung (REG-FE)
OB 35	Kommunikationsfehler auf der zweiten seriellen Schnittstelle
OB 36 bis 39	bei CPU 928 nicht vorhanden

1) Der OB 28 wird vor Übergang in den Stoppzustand aufgerufen. Der Stoppzustand erfolgt immer, gleichgültig ob und wie der OB 28 programmiert ist.

AUSNAHME: Bei NETZ AUS wird der OB 28 nicht aufgerufen.

Sonderfunktions Organisations-bausteine

Die folgenden Organisationsbausteine enthalten Sonderfunktionen des Systemprogramms. Sie können von Ihnen nicht programmiert (dies gilt für alle OBs mit Nummern zwischen 40 und 255), sondern lediglich aufgerufen werden. Sie enthalten kein Programm.

Sonderfunktions-OBs können in allen Code-Bausteinen aufgerufen werden.

Integrierte OBs mit Sonderfunktionen

Baustein	Bausteinfunktion
OB 110	Zugriff auf Anzeigenbyte
OB 111	AKKU 1, 2, 3 und 4 löschen
OB 112	AKKU Roll Up
OB 113	AKKU Roll Down
OB 120	"Alarme gemeinsam sperren" ein-/ausschalten
OB 121	"Weckalarme einzeln sperren" ein-/ausschalten
OB 122	"Alarme gemeinsam verzögern" ein-/ausschalten
OB 123	"Alarme gemeinsam verzögern" ein-/ausschalten
OB 134	*D
OB 135	/D
OB 136	MOD
OB 139	PUSH
OB 150	Systemzeit stellen/lesen
OB 151	Uhrzeitgesteuerte Weckalarmzeit stellen/lesen
OB 152	Zyklusstatistik
OB 160 bis 163	Zählschleifen
OB 170	Bausteinstack (BSTACK) lesen
OB 180	variabler Datenbaustein-Zugriff
OB 181	Datenbausteine DB/DX testen
OB 182	Datenbereich kopieren
OB 186	Komprimieren per Anwenderprogramm
OB 190, 192	Merker in Datenbaustein übertragen
OB 191, 193	Datenblöcke in Merkerbereich übertragen
OB 216 bis 218	Zugriffe auf "Kachel" (CPs und einige IPs)
OB 220	Vorzeichenerweiterung
OB 221	Zyklusüberwachungszeit einstellen
OB 222	Zyklusüberwachungszeit neu starten
OB 228	Statusinformation einer Programmbearbeitungsebene lesen
OB 230 bis 237	Funktionen für Standard-Funktionsbausteine (Hantierungsbausteine)
OB 250	Regelung: PID-Algorithmus initialisieren
OB 251	PID-Algorithmus bearbeiten
OB 254, 255	Datenbaustein ins DB-RAM übertragen (nicht 928C)

Die ausführliche Beschreibung dieser Sonderfunktionen finden Sie im Teil "Integrierte Sonderfunktionen".

Funktionsbausteine

Funktionsbausteine (FB/FX) sind ebenso Teile des Anwenderprogramms wie z.B. Programmbausteine. FX-Funktionsbausteine haben den gleichen Aufbau wie FB-Funktionsbausteine und werden ebenso programmiert.

Mit Funktionsbausteinen werden häufig wiederkehrende oder sehr komplexe Funktionen realisiert. Jeder Funktionsbaustein stellt innerhalb des Anwenderprogramms eine abgeschlossene Funktion dar. Sie können Funktionsbausteine

- als Softwareprodukt von VIPA beziehen mit diesen Standard-Funktionsbausteinen können Sie Ihre Programme für Steuern, Melden, Regeln und Protokollieren schnell und sicher erstellen.

  oder
- selbst programmieren.

Unterschiede zu den anderen Codebausteinen

Funktionsbausteine weisen gegenüber den Organisations-, Programm- und Schrittbausteinen vier wesentliche Unterschiede auf:

	OB, PB, SB	FB/FX
1.	Operationsumfang	
	nur Grundoperationen	Grundoperationen Ergänzende Operationen Systemoperationen
2.	Darstellungsart	
	Programmieren und Aufrufen in AWL, KOP, FUP	Programmieren nur in AWL
3.	Namen	
	Namensgebung nicht möglich (nur Nummer)	zusätzlich zur Nummer wird ein Name mit max. 8 Zeichen zugewiesen
4.	Operanden	
	keine	Formaloperanden (Bausteinparameter). Beim Aufruf werden den Formaloperanden Aktualoperanden zugewiesen.

Aufbau von Funktions-bausteinen

Bausteinkopf

Der Bausteinkopf (5 Wörter) eines Funktionsbausteins ist gleich aufgebaut wie die Bausteinköpfe der übrigen Bausteine.

Bausteinrumpf

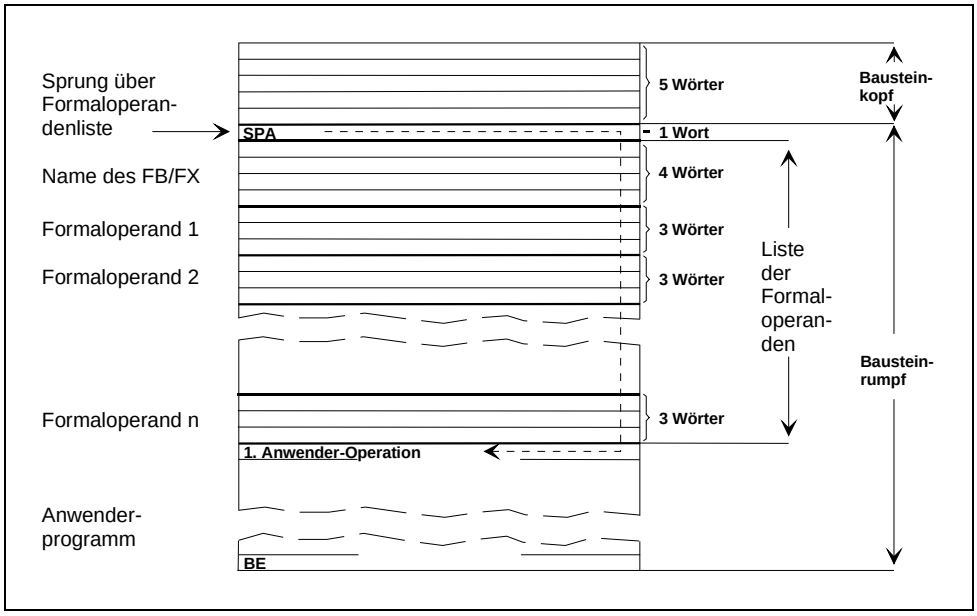
Der Bausteinrumpf hingegen unterscheidet sich in seinem Aufbau von dem der anderen Bausteinarten. Die auszuführende Funktion ist darin in Form einer Anweisungsliste in der Programmiersprache geschrieben. Zwischen dem Bausteinkopf und den Anweisungen benötigt ein Funktionsbaustein weiteren Speicherplatz für die Angabe seines Namens und für die Liste der Formaloperanden. Da diese Liste keine Anweisungen für die CPU enthält, wird sie mit einem unbedingten Sprung übersprungen, den das PG automatisch erzeugt. Diese Sprunganweisung wird bei der Ausgabe am PG nicht angezeigt! Bei einem Aufruf des Funktionsbausteins wird nur der Bausteinrumpf bearbeitet.

Absolute oder symbolische Operanden

Operanden können in einem Funktionsbaustein absolut (z.B. M 2.5) oder symbolisch (z.B. Motor1) eingegeben werden. Die Zuordnung der symbolischen Operanden müssen Sie zuvor in einer Zuordnungsliste ablegen.

Funktionsbaustein im AG-Speicher

So sieht ein Funktionsbaustein im AG-Speicher aus:



Im Speicher stehen somit alle Angaben, die das Programmiergerät benötigt, um den Funktionsbaustein beim Aufruf grafisch darstellen zu können und um die Operanden bei der Parametrierung und Programmierung des Funktionsbausteines zu überprüfen. Eine fehlerhafte Eingabe wird vom Programmiergerät abgelehnt.

Unterschied: "Programmieren" - "Aufrufen und Parametrieren"

Unterscheiden Sie bei der Behandlung von Funktionsbausteinen zwischen:

- FB/FX programmieren
- FB/FX aufrufen und anschließend parametrieren.

Programmieren

Beim Programmieren legen Sie die Funktion des Bausteins fest. Hierbei müssen Sie überlegen, welche Eingangsoperanden die Funktion benötigt und welche Ausgangsergebnisse sie an das aufrufende Programm übergeben soll. Alle Eingangsoperanden und Ausgangsergebnisse definieren Sie dann als Formaloperanden. Diese üben eine Platzhalterfunktion aus.

Aufrufen und Parametrieren

Beim Aufrufen eines Bausteins durch einen übergeordneten Baustein (OB, PB, SB, FB, FX) werden die Formaloperanden (Bausteinparameter) durch Aktualoperanden ersetzt: der Funktionsbaustein wird parametriert.

Vorgehen beim Programmieren

WENN...	DANN...
Sie einen Funktionsbaustein "direkt", d.h. ohne Formaloperanden programmieren wollen,	führen Sie dies durch wie bei Programm- oder Schrittbausteinen.
Sie in einem Funktionsbaustein Formaloperanden verwenden wollen,	gehen Sie so vor, wie auf den nächsten Seiten beschrieben. Beachten Sie dabei die erforderliche Reihenfolge: den FB/FX mit den Formaloperanden programmieren und im PG (offline) oder im CPU-Speicher (online) halten, den/die aufrufenden Baustein(e) mit den Aktualoperanden programmieren.

Programmieren von Funktionsbausteinen

Einen Funktionsbaustein können Sie nur in der Darstellungsart "Anweisungsliste" programmieren. Bei der Eingabe eines Funktionsbausteines am PG gehen Sie so vor:

1. Geben Sie den Bausteintyp (FB/FX) und Nummer des Funktionsbausteins ein.

   Anwender-Funktionsbausteine sollten von FB 255 an fallend numeriert werden, um nicht mit den Standard-Funktionsbausteinen zu kollidieren, die von FB 1 bis FB 199 numeriert sind.

2. Geben Sie den Namen des Funktionsbausteins ein.

   Der Name kann bis zu 8 Zeichen lang sein und muss mit einem Buchstaben beginnen.

3. Wenn der Funktionsbaustein Formaloperanden bearbeiten soll:

   Geben Sie als Bausteinparameter die Formaloperanden ein, die Sie im Baustein verwenden.

   Für jeden Formaloperanden müssen Sie angeben:

   den Namen des Bausteinparameters (maximal 4 Zeichen),

   die Art des Bausteinparameters und evtl. den Typ des Bausteinparameters.

   Sie können maximal 40 Formaloperanden definieren.

4. Geben Sie Ihr Programm in der Darstellungsart AWL ein. Die Formaloperanden werden dabei durch ein vorangestelltes Gleichheitszeichen gekennzeichnet (z.B. U=X1). Sie können auch mehrmals an verschiedenen Stellen im Funktionsbaustein angesprochen werden.

5. Schließen Sie die Programmeingabe mit "BE" (Bausteinende) ab.

Hinweis!

Wenn Sie die Reihenfolge oder die Anzahl der Formaloperanden in der Formaloperandenliste ändern, müssen alle Anweisungen im Funktionsbaustein, die einen Formaloperanden ansprechen, und die Bausteinparameterliste im aufrufenden Baustein entsprechend nachgeführt werden!

Programmieren und ändern Sie Funktionsbausteine grundsätzlich auf Diskette oder Festplatte und übertragen Sie sie anschließend in die CPU!

Formaloperanden

Als Formaloperanden eines Funktionsbausteins (auch Bausteinparameter genannt) sind folgende Art- und Typbezeichnungen zugelassen.

Parameterart	Parametertyp
E = Eingang A = Ausgang	BI/BY/W/D
D = Datum	KM/KH/KY/KC/KF/KT/KZ/KG
B = Befehl T = Zeit (Timer) Z = Zähler	keine (Typangabe unzulässig)

E, D, B, T oder Z sind Parameterarten, die bei der graphischen Darstellung eines FB-Aufrufs auf der linken Seite des Funktionssymbols gezeichnet werden.

Mit A gekennzeichnete Parameter werden bei der graphischen Darstellung eines FB-Aufrufs auf der rechten Seite des Funktionssymbols gezeichnet.

Der Parametertyp gibt an, ob es sich bei E- und A-Parametern um Bit-, Byte-, Wort- oder Doppelwortgrößen handelt und welches Datenformat (z.B. Bitmuster oder Hexadezimalmuster) für D-Parameter gilt.

Aufruf und Parametrierung von Funktions-bausteinen

Jeder Funktionsbaustein kann beliebig oft und an beliebigen Stellen im Anwenderprogramm aufgerufen werden. Aufrufe von Funktionsbausteinen können sowohl in einer Anweisungsliste als auch in einer graphischen Darstellung erfolgen (FUP oder KOP).

Gehen Sie beim Aufrufen und Parametrieren folgendermaßen vor:

1. Stellen Sie sicher, dass der aufgerufene Funktionsbaustein entweder im PG-Speicher (offline) oder im CPU-Speicher (online) vorhanden ist.

2. Geben Sie im aufrufenden Baustein die Aufrufanweisung für den Funktionsbaustein ein.

   Der Aufruf eines Funktionsbausteins kann innerhalb eines Organisations-, Programm- oder Schrittbausteins oder innerhalb eines anderen Funktionsbaustein programmiert werden.

   Reaktion am PG:

   Nachdem Sie die Aufrufanweisung (z.B. SPA FB 200) eingegeben haben, erscheinen automatisch der Name und die Formaloperandenliste des betreffenden Funktionsbausteins.

3. Ordnen Sie jedem einzelnen Formaloperanden den für diesen Aufruf gültigen Aktual-Operanden zu. (Sie parametrieren dadurch den Funktionsbaustein).

   Die Aktualoperanden können bei den einzelnen Aufrufen unterschiedlich sein: beim ersten Aufruf des FB 200 z.B. Ein- und Ausgänge, beim zweiten Aufruf Merker.

   Entsprechend der Formaloperandenliste müssen Sie bei jedem Aufruf eines Funktionsbausteins die erforderlichen Aktualoperanden zuordnen.

Aufruf unbedingt/bedingt

unbedingter Aufruf	bedingter Aufruf
"SPA FBn" für Funktionsbausteine FB oder "BA FXn" für erweiterte Funktionsbausteine FX: Der angesprochene Funktionsbaustein wird unabhängig vom Verknüpfungsergebnis (VKE) bearbeitet.	"SPB FBn" für Funktionsbausteine FB oder "BAB FXn" für erweiterte Funktionsbausteine FX: Der angesprochene Funktionsbaustein wird nur dann bearbeitet, wenn das Verknüpfungsergebnis VKE=1 ist. Bei VKE=0 wird die Sprunganweisung nicht ausgeführt. Unabhängig davon, ob der Bausteinaufruf ausgeführt wird oder nicht, wird das VKE immer auf '1' gesetzt.
Nach dem unbedingten und bedingten Aufruf kann das Verknüpfungsergebnis nicht weiter verknüpft werden. Es bleibt beim Sprung in den FB erhalten und kann dort ausgewertet werden.	

Zulässige Aktualoperanden

Welche Operanden Sie einem Funktionsbaustein zuordnen dürfen, entnehmen Sie bitte der nachfolgenden Tabelle:

Hinweis!

S-Merker sind als Aktualoperanden für Funktionsbausteine nicht zugelassen.

Art des Parameters	Typ des Parameters		Zugelassene Aktualoperanden		
E, A	BI	für einen Operanden mit Bitadresse	E A M	n.m n.m n.m	Eingang Ausgang Merker
	BY	für einen Operanden mit Byteadresse	EB AB MB DL DR PY QB	n n n n n n n	Eingangsbyte Ausgangsbyte Merkerbyte Datenbyte links Datenbyte rechts Peripheriebyte Byte der erweiterten Peripherie
	W	für einen Operanden mit Wortadresse	EW AW MW DW PW QW	n n n n n n	Eingangswort Ausgangswort Merkerwort Datenwort Peripheriewort Wort der erweiterten Peripherie
	D	für einen Operanden mit Doppelwortadresse	ED AD MD DD	n n n n	Eingangs-Doppelwort Ausgangs-Doppelwort Merker-Doppelwort Daten-Doppelwort
D	KM	für ein Binärmuster (16 Stellen)	Konstante		
	KY	für zwei byteweise Betragszahlen im Bereich von jeweils 0 bis 255			
	KH	für ein Hexadezimalmuster bis 4 Stellen			

Fortsetzung ...

... Fortsetzung

Art des Parameters	Typ des Parameters		Zugelassene Aktualoperanden		
D	KC	für zwei alphanumerische Zeichen	Konstante		
	KT	für einen Zeitwert (BCD-codiert) mit Zeitraster .0 bis .3 und Zeitwert 0 bis 999			
	KZ	für einen Zählwert 0 bis 999			
	KF	für eine Festpunktzahl -32768 bis +32767			
	KG	für eine Gleitpunktzahl $\pm 0{,}1469368 \times 10^{-38}$ bis $\pm 0{,}1701412 \times 10^{39}$			
B	keine Typangabe zulässig		DB	n	Datenbaustein; ausgeführt wird der Befehl A DB n
			FB	n	Funktionsbaustein (nur ohne Parameter zulässig) wird unbedingt (SPA ..n) aufgerufen
			OB	n	Organisationsbaustein; wird unbedingt (SPA ..n) aufgerufen
			PB	n	Programmbaustein; wird unbedingt (SPA ..n) aufgerufen
			SB	n	Schrittbaustein; wird unbedingt (SPA ..n) aufgerufen
T	keine Typangabe zulässig		T 0 bis 255 Zeit		
Z	keine Typangabe zulässig		Z 0 bis 255 Zähler		

Nach dem Sprung in den Funktionsbaustein werden bei der Bearbeitung des Funktionsbaustein-Programms anstelle der Formaloperanden die Aktualoperanden aus dem aufrufenden Baustein verwendet.

Aufgrund dieser Eigenschaft der parametrierbaren Funktionsbausteine können Sie diese in Ihrem Anwenderprogramm vielseitig verwenden.

Beispiele

Beispiel 1

Folgendes (unvollständiges) Beispiel soll Ihnen sowohl das Programmieren als auch das Aufrufen und Parametrieren eines Funktionsbausteins verdeutlichen. Sie können es selbst leicht nachvollziehen.

Der Funktionsbaustein FB 202 wird programmiert:

FB 202

```
NETZWERK 1
NAME      BEISPIEL
BEZ             MONI   E/A/D/B/T/Z:   E     E/A/D/B/T/Z:BI
BEZ             BERT   E/A/D/B/T/Z:   E     E/A/D/B/T/Z:BI
BEZ             HANS   E/A/D/B/T/Z:   A     E/A/D/B/T/Z:BI
          :U=   MONI
          :U=   BERT
          :==   HANS
          :
          : BE
```

Formaloperandenliste

Anweisungen

Formaloperanden | Parameterart | Parametertyp

Der Funktionsbaustein FB 202 wird im Programmbaustein PB 25 aufgerufen und parametriert:

Darstellungsart AWL

```
PB 25
NETZWERK 1
          :SPA FB 202
NAME      :BEISPIEL
MONI      :E   13.5
BERT      :M   17.7
HANS      :A   23.0
          :BE
```

Formal-operanden | Aktual-operanden

Darstellungsart KOP/FUP

Folgende Operationen werden nach Sprung in den FB 202 ausgeführt:

```
:U    E     13.5
:U    M     17.7
:=    A     23.0
:BE
```

Beispiel 2

Aufruf und Parametrierung eines Funktionsbausteins mit den Darstellungsarten AWL und KOP/FUP in einem Programmbaustein.

Darstellungsart AWL

```
FB 25
NETZWERK 1
      :A        DB      5
      :SPA      FB      201
NAME:E-ANTR
ZU-E :  DW              1
RME  :  E               3.5
ESB  :  M               2.5
UEZ  :  T               2
ZEIT :  KT              010.1
ZU-A :  DW              2
BEA  :  A               2.3
LSL  :  A               6.0
      :BE
```

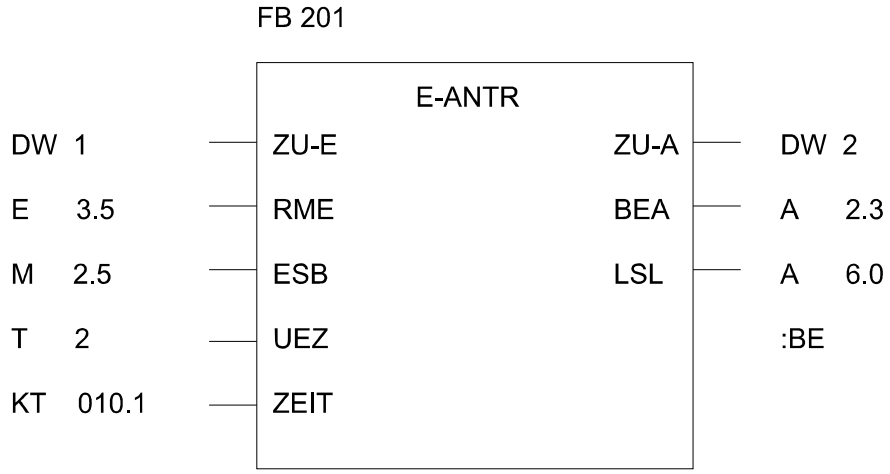
Formal-operanden

Aktual-operanden

Darstellungsart KOP/FUP

PB 25

NETZWERK 1



Standard-Funktions-bausteine

Neben den Funktionsbausteinen, die Sie selbst programmieren, bietet Ihnen die VIPA Standard-Funktionsbausteine als fertiges Softwareprodukt (Best.-Nr.: VIPA-SW928). Standard-Funktionsbausteine belegen die Nummern FB 120 bis FB 127.

Datenbausteine

In einem Datenbaustein (DB) oder einem erweiterten Datenbaustein (DX) sind die festen oder variablen Daten abgelegt, mit denen Ihr Anwenderprogramm arbeitet. In Datenbausteinen werden keine Operationen bearbeitet.

Die Daten eines Datenbausteins können sein:

- beliebige Bitmuster, z.B. für Anlagenzustände
- Zahlen (hexadezimal, dual, dezimal) für Zeitwerte, Rechenergebnisse
- alphanumerische Zeichen, z.B. für Meldetexte

Aufbau eines Datenbausteins

Ein Datenbaustein (DB/DX) besteht aus den Teilen

- Bausteinvorkopf (DV, DXV)
- Bausteinkopf
- Bausteinrumpf.

Bausteinvorkopf

Der Bausteinvorkopf wird automatisch auf der Festplatte oder der Diskette des PG angelegt und nicht in die CPU übertragen. Er enthält die Datenformate der im Bausteinrumpf eingegebenen Datenwörter.

Als Anwender haben Sie keinen Einfluss auf das Anlegen des Bausteinvorkopfes.

Achtung!

Wenn Sie einen Datenbaustein aus der CPU auf Diskette oder auf Festplatte übertragen, wird der dazugehörige Bausteinvorkopf gelöscht. Aus diesem Grund dürfen Sie einen Datenbaustein mit unterschiedlichen Datenformaten nie in der CPU ändern und ihn anschließend zurück übertragen, sonst wird allen Datenwörtern dieses DB automatisch das Datenformat zugeordnet, das Sie in der Voreinstellungsmaske gewählt haben.

Bausteinkopf

Der Bausteinkopf belegt 5 Wörter im Speicher und enthält

- die Bausteinkennung,
- die Kennung des Programmiergerätes,
- den Bausteintyp und die Bausteinnummer,
- die Bibliotheksnummer
- die Bausteinlänge (inkl. Länge des Bausteinkopfes).

Bausteinrumpf

Der Bausteinrumpf enthält in aufsteigender Reihenfolge, beginnend mit Datenwort DW 0, die Datenwörter, mit denen das Anwenderprogramm arbeitet. Jedes Datenwort belegt im Speicher 1 Wort (16 Bit).

Maximale Länge

Datenbausteine dürfen insgesamt bis zu 4096 Wörter (inkl. Kopf) im Speicher der CPU belegen. Berücksichtigen Sie beim Eingeben und Übertragen von Datenbausteinen mit dem PG den Speicherausbau Ihrer CPU!

Erstellen von Datenbausteinen

So erstellen Sie einen Datenbaustein:

1. Geben Sie den Bausteintyp (DB/DX) und eine Datenbaustein-Nummer zwischen 3 und 255 ein.
2. Geben Sie die einzelnen Datenwörter im gewünschten Datenformat ein.

(Die Eingabe der Datenwörter wird nicht mit einer BE-Anweisung abgeschlossen.)

Achtung!

Die Datenbausteine DB 0, DB 1, DB 2, DX 0, DX 1 und DX 2 sind für bestimmte Funktionen reserviert und damit nicht frei verwendbar.

Zulässige Datenformate

In einem Datenbaustein können Sie die in der folgenden Tabelle aufgeführten Datenformate verwenden.

Bezeichnung	Datenformat	Beispiel
KM	Bitmuster	00100110 00111111
KH	Hexadezimalzahl	263F
KY	Byte	038,063
KF	Festpunktzahl	09791
KG	Gleitpunktzahl	+1356123+12
KC	Zeichen	?!ABCD123-+.,%
KT	Zeitwert eines Zeitglieds	055.2
KZ	Zählerwert	234

Aufschlagen von Datenbausteinen

Ein Datenbaustein (DB/DX) kann nur unbedingt aufgeschlagen werden. Dies ist möglich innerhalb eines Organisations-, Programm-, Schritt- oder Funktionsbaustein. Ein bestimmter Datenbaustein kann mehrfach im Programm aufgeschlagen werden.

So schlagen Sie Datenbausteine auf:

WENN...	DANN...
Sie einen DB-Datenbaustein aufschlagen wollen,	geben Sie die Operation "A DB.." ein.
Sie einen DX-Datenbaustein aufschlagen wollen,	geben Sie die Operation "AX DX.." ein.

Gültigkeitsbereich

Nach Aufschlagen eines Datenbausteins beziehen sich alle folgenden Anweisungen mit dem Operandenbereich **'D'** auf den aufgeschlagenen Baustein.

Der aufgeschlagene Datenbaustein bleibt auch dann gültig, wenn durch einen Bausteinaufruf die Programmbearbeitung in einem anderen Baustein fortgesetzt wird.

Wenn in diesem Baustein nun ein anderer Datenbaustein aufgeschlagen wird, ist dieser **nur** im aufgerufenen Baustein gültig. Nach Rücksprung in den aufrufenden Baustein gilt wieder der alte Datenbaustein.

Zugriff

Der Zugriff auf die in dem aufgeschlagenen Datenbaustein gespeicherten Daten erfolgt bei der Programmbearbeitung durch die Lade- und Transferoperationen.

Mit einer *binären Verknüpfung* wird das adressierte Datenwort-Bit zur Bildung des VKE herangezogen. Der Inhalt des Datenwortes wird nicht verändert.

Bei einer *Speicheroperation* wird dem adressierten Datenwort-Bit der Wert des VKE zugewiesen. Der Inhalt des Datenwortes kann dabei verändert werden.

Mit einer *Ladeoperation* wird der Inhalt des adressierten Datenwortes in den AKKU 1 übertragen. Der Inhalt eines Datenwortes wird nicht verändert.

Mit einer *Transferoperation* werden die Daten aus dem AKKU 1 in das adressierte Datenwort übertragen. Der alte Inhalt eines Datenwortes wird überschrieben.

Hinweis!

Vor dem Zugriff auf ein Datenwort müssen Sie im Anwenderprogramm den gewünschten Datenbaustein aufschlagen, da die CPU nur so das richtige Datenwort findet.

Das adressierte Datenwort muss im aufgeschlagenen Baustein enthalten sein, sonst erkennt das Systemprogramm bei einem Zugriff einen Lade- bzw. Transferfehler.

Mit Lade- und Transferoperationen können Sie nur bis zu Datenwortnummer 255 zugreifen!

Ein aufgeschlagener Datenbaustein bleibt gültig, bis

- ein anderer Datenbaustein aufgeschlagen wird

  oder
- ein aufgerufener Baustein mit 'BE' oder 'BEB' beendet wird.

Beispiele

Beispiel 1: Transferieren von Datenwörtern

Es soll der Inhalt des Datenwortes DW 1 vom Datenbaustein DB 10 in das Datenwort DW 1 des Datenbausteins DB 20 transferiert werden.

Dazu geben Sie folgende Anweisungen ein:

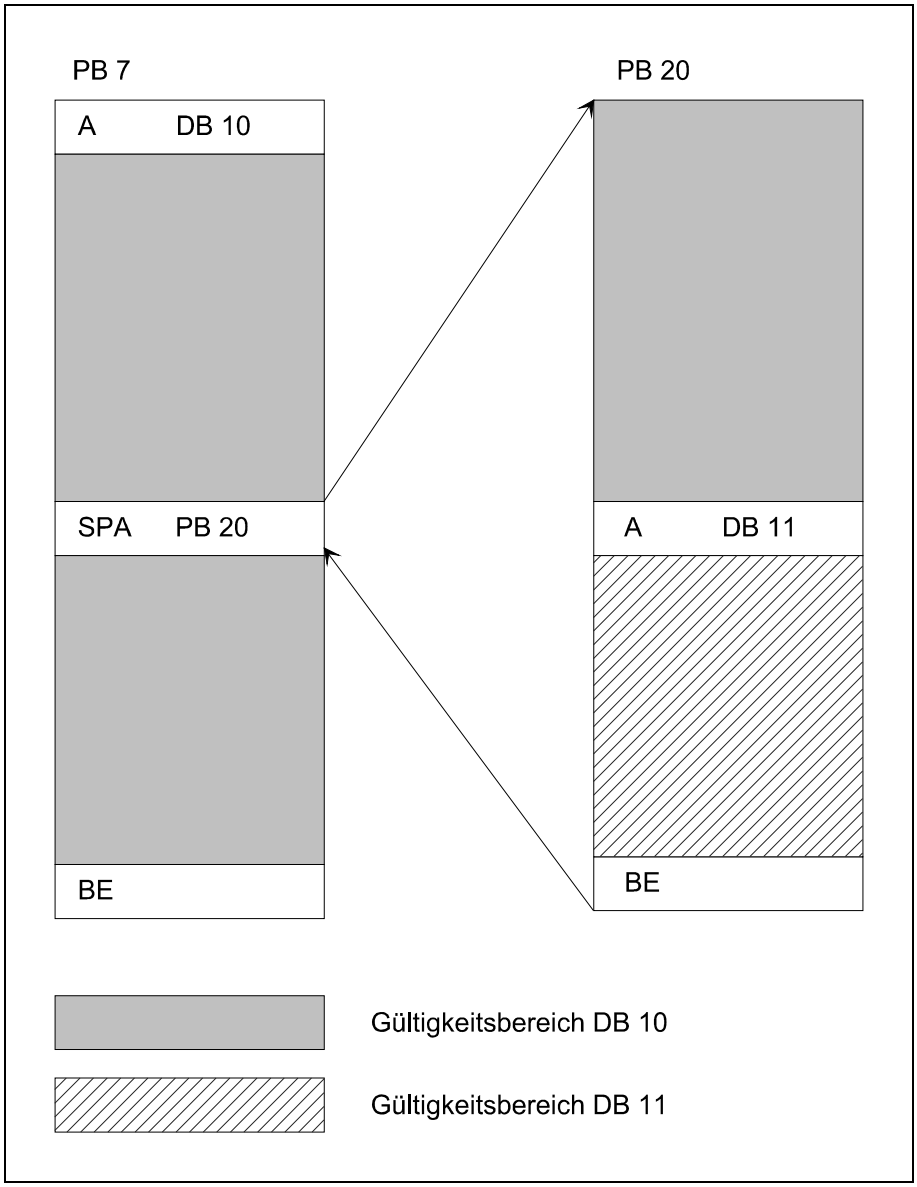
```
:A    DB   10   (DB 10 aufschlagen)
:L    DW   1    (Inhalt DW 1 in den
:               AKKU 1 übertragen)
:A    DB   20   (DB 20 aufschlagen)
:T    DW   1    (Inhalt AKKU 1 nach
:               DW 1 übertragen)
:
```

Beispiel 2: Gültigkeitsbereich bei Datenbausteinen

Im Programmbaustein PB 7 wird der Datenbaustein DB 10 aufgeschlagen (A DB 10). In der folgenden Programmbearbeitung werden die Daten dieses Datenbausteins bearbeitet.

Nach dem Aufruf (SPA PB 20) wird der Programmbaustein PB 20 bearbeitet. Der Datenbaustein DB 10 ist jedoch nach wie vor gültig. Erst mit dem Aufschlagen des Datenbausteins DB 11 (A DB 11) wird der Datenbereich gewechselt. Bis zum Ende des Programmbausteins PB 20 (BE) ist nun der Datenbaustein DB 11 gültig.

Nach der Rückkehr in den Programmbaustein PB 7 ist wieder der Datenbaustein DB 10 gültig.



Spezielle Datenbausteine

Die Datenbausteine DB 0, DB 1, DX 0, DX 1 und DX 2 sind bei der CPU 928 für bestimmte Funktionen reserviert. Sie werden vom Systemprogramm verwaltet und sind für den Anwender nicht beliebig verwendbar.

DB 0

(siehe Teil "Speicherbelegung und Organisation")**:**

Der Datenbaustein DB 0 enthält die Adressliste mit den Anfangsadressen aller Bausteine, die sich im Anwenderspeicher oder im Datenbaustein-RAM der CPU befinden. Diese Adressliste wird vom Systemprogramm bei der Initialisierung (nach URLÖSCHEN) erzeugt und bei der Eingabe oder Änderung von Bausteinen mit dem PG automatisch aktualisiert.

DB 1

Der Datenbaustein DB 1 enthält die Liste der digitalen Ein- und Ausgänge (P-Peripherie mit relativen Byteadressen von 0 bis 127), die der CPU zugeordnet sind, und gegebenenfalls eine Zeitenblocklänge.

- DB 1 **kann** parametriert und geladen werden:

  Um die Zykluszeit im Einzelprozessorbetrieb zu verringern, da nur die im DB 1 eingetragenen Ein- und Ausgänge oder Zeiten aktualisiert werden.

DX 0

(siehe Teil "Erweiterter Datenbaustein DX 0"):

Durch Parametrieren und Laden des Datenbausteins DX 0 können Sie die Voreinstellungen bestimmter Systemprogrammfunktionen (z.B. bei der Bearbeitung des Anlaufs) ändern und damit die Leistungen des Systemprogramms Ihren Erfordernissen anpassen.

DX 1

Reserviert.

DX 2

Der Datenbaustein DX 2 wird dazu verwendet, um für die Kommunikation über die 2. serielle Schnittstelle den Kopplungstyp festzulegen.

Teil 4 Programmbearbeitung

Überblick

Hier finden Sie eine Einführung in die Programmiersprache. Es werden im Weiteren die Operationen anhand von Beispielen erläutert.

Nachfolgend sind beschrieben:

- Programmbearbeitung und Programmorganisation
- Speicherung von Programm- und Datenbausteinen
- Bearbeitung und Begriffsdefinition des Anwenderprogramms
- Grundoperationen
- Beispiele in AWL, KOP, und FUP
- Ergänzende Operationen
- Organisatorische Operationen
- Semaphor-Operationen

Inhalt

Thema **Seite**

Teil 4 Programmbearbeitung....4-1

Prinzip der Programmbearbeitung....4-2

Programmorganisation....4-3

Speicherung von Programm- und Datenbausteinen....4-8

Bearbeitung des Anwenderprogramms....4-10

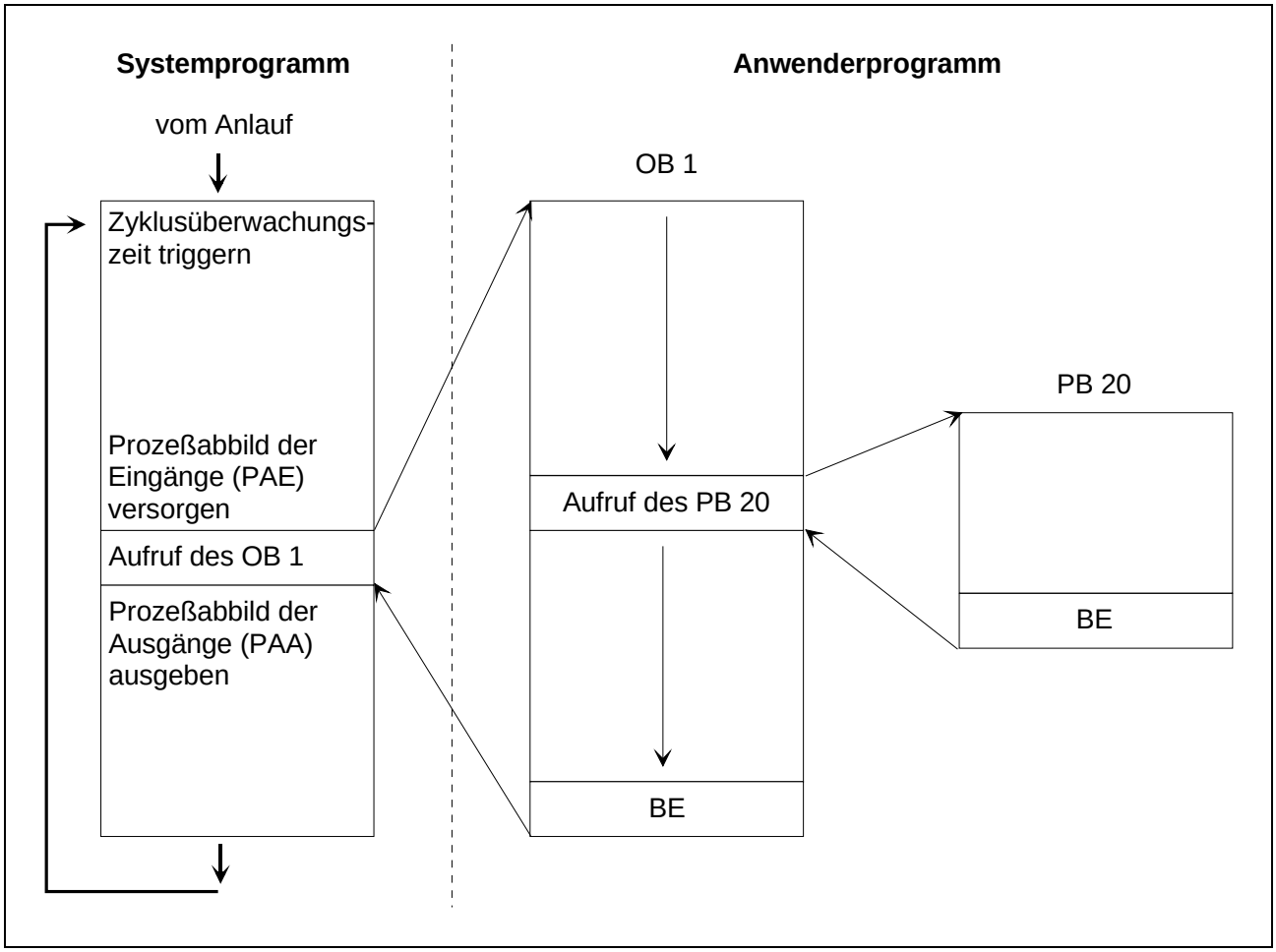
Operationen mit Beispielen....4-13

Prinzip der Programmbearbeitung

Das Anwenderprogramm kann auf verschiedene Art und Weise bearbeitet werden.

Typischerweise herrscht bei speicherprogrammierbaren Steuerungen (SPS) die zyklische Programmbearbeitung vor:

Das Systemprogramm läuft in einer Programmschleife (dem Zyklus) und ruft dabei in jeder Schleife einmal den Organisationsbaustein OB 1 auf .



Programmorganisation

Einleitung

Mit der Programmorganisation legen Sie fest, nach welchen Bedingungen und in welcher Reihenfolge die von Ihnen erstellten Bausteine bearbeitet werden sollen.

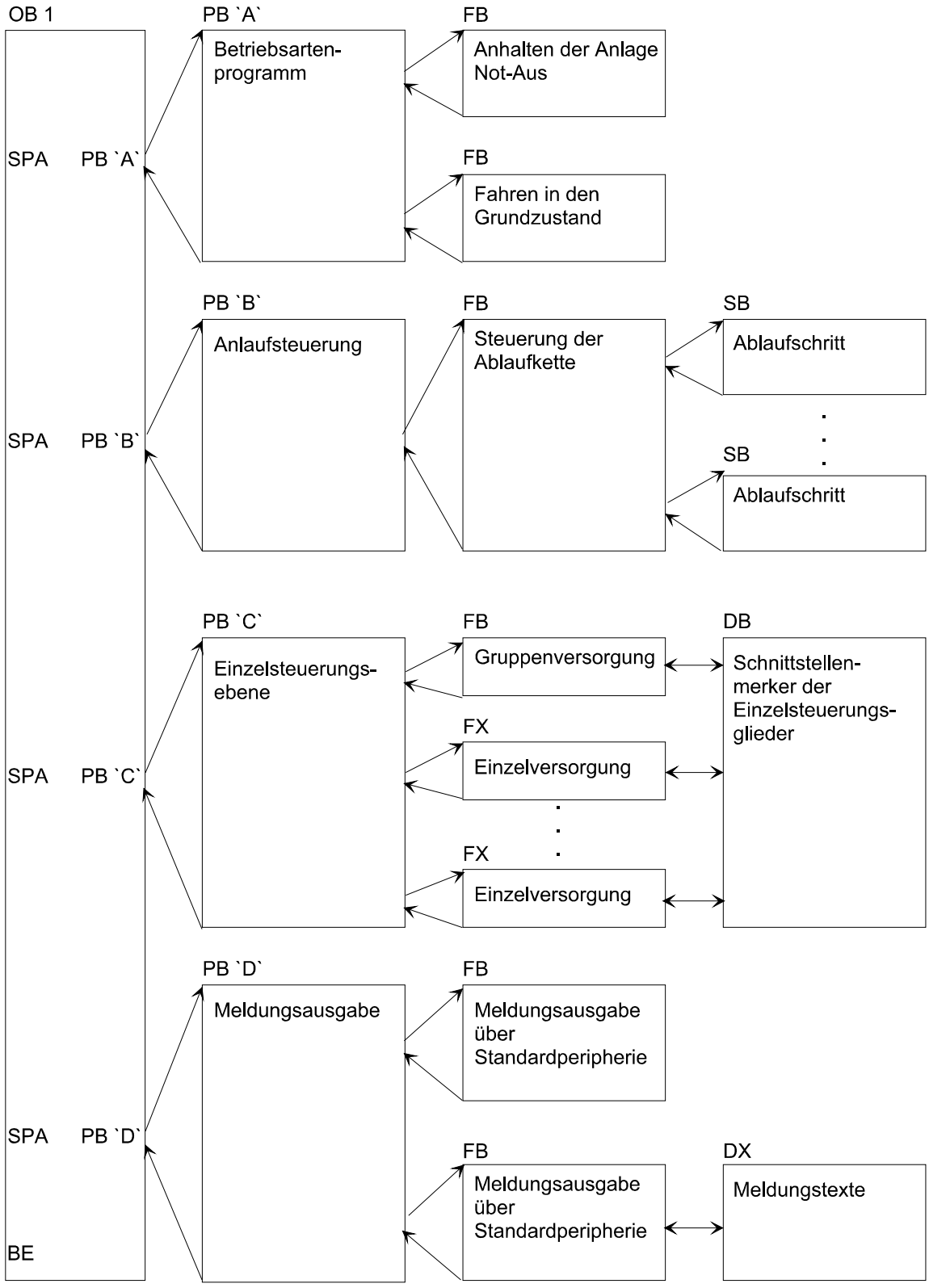
Dazu programmieren Sie in den Organisationsbausteinen bedingte oder unbedingte Aufrufe der gewünschten Bausteine.

In den Programmteilen der einzelnen Organisations-, Programm-, Funktions- und Schrittbausteine können weitere Programm-, Funktions- und Schrittbausteine in beliebiger Kombination (nacheinander oder ineinander verschachtelt) aufgerufen werden.

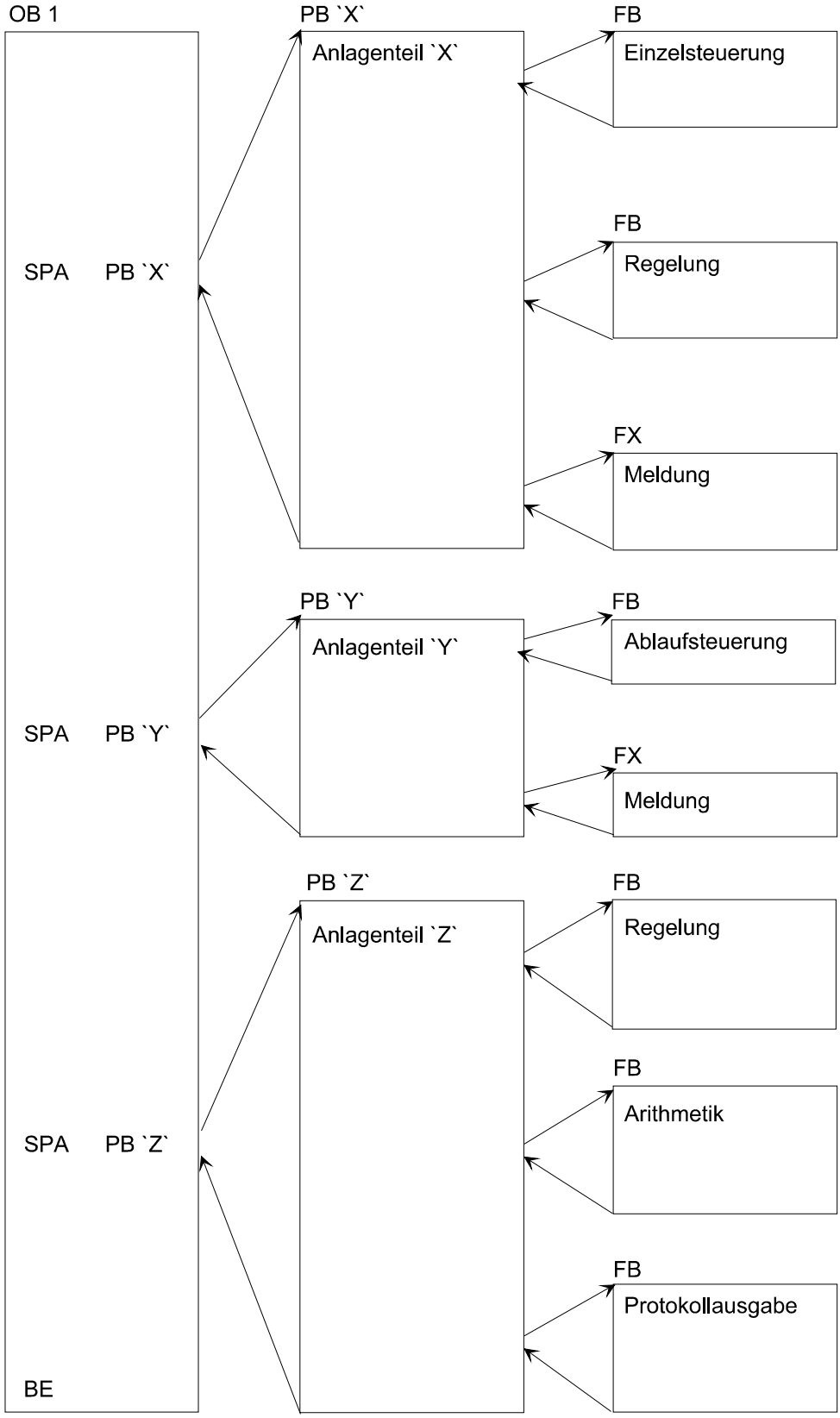
Das Anwenderprogramm sollte zweckmäßigerweise so organisiert sein, dass es die wesentlichen Programmstrukturen oder programmtechnisch zusammenhängende Anlagenteile hervorhebt.

Die folgenden Abbildungen und zeigen Ihnen zwei Beispiele einer Programmstruktur.

**Programm-
organisation nach
Programmstruktur**

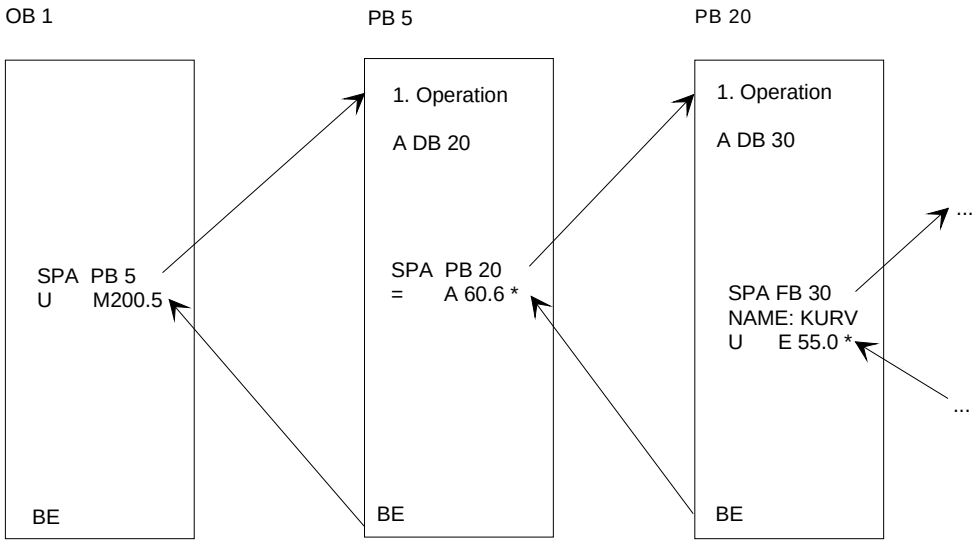


**Programm-
organisation nach
Anlagenstruktur**



Baustein-schachtelung

Das Bild zeigt Ihnen das Prinzip der verschachtelten Bausteinaufrufe.



* Operation zu der zurückgesprungen wird

Bausteinadressen

Die Lage eines Bausteins im Anwenderspeicher (oder DB-RAM) ist festgelegt durch seine Baustein-Anfangsadresse: Dies ist bei Code-Bausteinen die Adresse derjenigen Zelle im Speicher, in der sich die erste Operation des Bausteins befindet (bei FB und FX der SPA-Befehl über die Formaloperandenliste); bei Datenbausteinen ist es die Adresse des ersten Datenwortes.

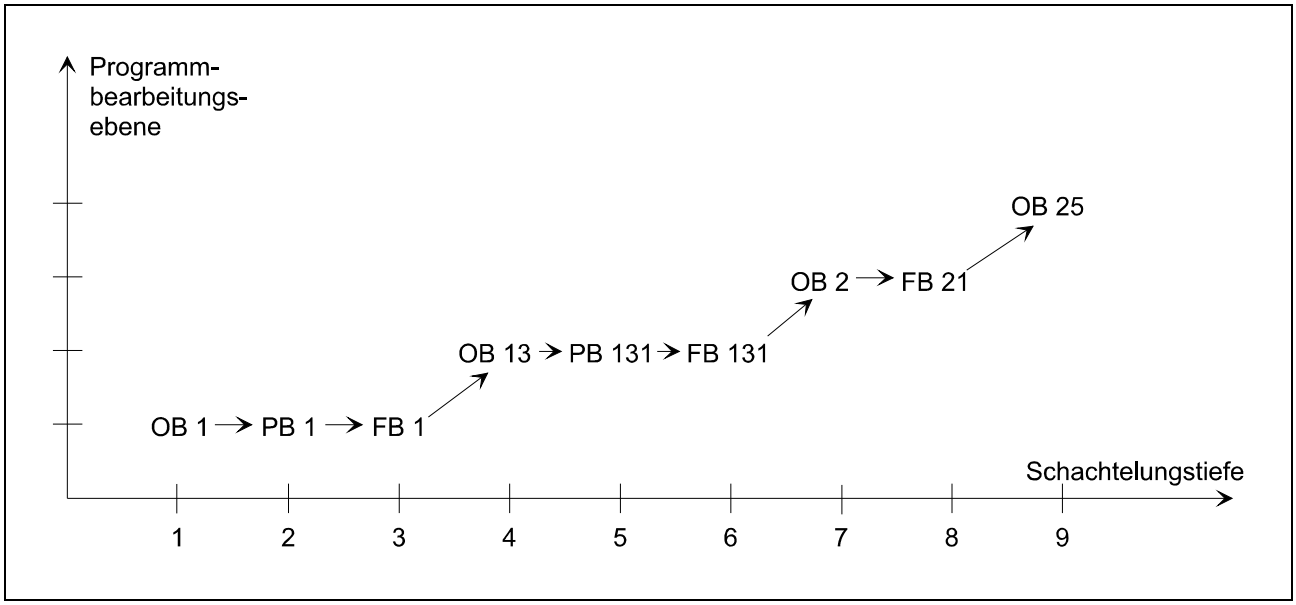
Damit die CPU bei einem Bausteinaufruf den aufgerufenen Baustein im Speicher findet, sind die Anfangsadressen aller gültigen Bausteine in der Bausteinadressliste im Datenbaustein DB 0 eingetragen. Der DB 0 wird vom Systemprogramm verwaltet, als Anwender können Sie ihn nicht aufschlagen!

Um nach Abarbeitung des aufgerufenen Bausteins den Rückweg in den aufrufenden Baustein zu finden, speichert die CPU bei jedem Aufruf eines neuen Bausteins die Rücksprungadresse: Die Rücksprungadresse ist die Adresse derjenigen Zelle im Speicher, in der die dem Bausteinaufruf folgende Anweisung steht. Außerdem wird von der CPU die Anfangsadresse und Länge des Datenbausteins gespeichert, der an dieser Stelle gültig ist.

Schachteltiefe

Sie können maximal 62 Bausteine ineinander schachteln. Werden mehr als 62 Bausteine aufgerufen, meldet die CPU einen Fehler und geht in den Stoppzustand.

Beispiel



So ermitteln Sie die Schachtelungstiefe Ihres Programms:

- Addieren Sie alle von Ihnen programmierten Organisationsbausteine (im Beispiel: 4 OBs).

- Addieren Sie die Schachtelungstiefen der einzelnen OBs (im Beispiel: 2 + 2 +1+ 0 = 5)

  (PB 1, FB 1) (PB 131, FB 131) (FB 21)

- Beide Beträge zusammen ergeben die Programm-Schachtelungstiefe im Beispiel: 4 + 5 = 9 Schachtelungstiefe.

  (OBs) (PBs und FBs)

Hinweis!
Die Schachtelungstiefe darf den Wert 62 nicht überschreiten.

Speicherung von Programm- und Datenbausteinen

Damit die CPU das Anwenderprogramm bearbeiten kann, muss dieses - einschließlich vorhandener Datenbausteine - in den Programmspeicher geladen werden. Als Programmspeicher stehen Ihnen Anwenderspeicher und das DB-RAM zur Verfügung.

Um Code- und Datenbausteine in den Programmspeicher zu laden, können Sie folgende Methoden benutzen:

RAM-Betrieb

Wenn beim URLÖSCHEN der CPU keine Memory Card (Flash EPROM) gesteckt ist, geht die CPU in den "RAM-Betrieb":

Code- und Datenbausteine werden vom PG in den Arbeitsspeicher bzw. in das DB-RAM der CPU geladen. Sie können jederzeit vom PG nachgeladen (ersetzt), verändert oder gelöscht werden, d.h. der Schreibschutz ist ausgeschaltet.

EPROM-Betrieb

Code- und Datenbausteine werden durch das Systemprogramm von der Memory Card in den Anwenderspeicher kopiert. Das Systemprogramm setzt außerdem eine Schreibschutzkennung.

Dies bedeutet, dass alle kopierten Bausteine nicht nachgeladen, verändert oder gelöscht werden können.

Um Daten in Datenbausteinen ändern zu können, müssen Sie dafür sorgen, dass die Datenbausteine ins DB-RAM gelangen.

Datenbausteine, die in der Memory Card programmiert sind, können Sie mit Hilfe der OBs 254/255 ins DB-RAM kopieren oder verschieben (z.B. im Anlauf-OB). Andere Datenbausteine können Sie vom PG aus ins DB-RAM laden. Die CPU 928C legt den DB automatisch im DB-RAM ab.

Sobald die Datenbausteine im DB-RAM liegen, können Sie nachgeladen, geändert oder gelöscht werden.

Änderungen an diesen Datenbausteinen werden jedoch nicht in die Memory Card übernommen. Vor dem nächsten Urlöschen müssen Sie den Inhalt der Memory Card retten.

Nach dem Urlöschen können Sie die Memory Card ziehen, die Daten stehen im Anwenderspeicher bereit.

Ein-/Ausschalten des Schreibschutzes

Falls Sie im EPROM-Betrieb Änderungen an Bausteinen vornehmen müssen, so können Sie den Schreibschutz durch Löschen der Schreibschutzkennung wieder aufheben.

Sie können Bausteine wieder mit einem Schreibschutz versehen, wenn Sie die Schreibschutzkennung setzen.

Sie können ebenso eine CPU ohne Memory Card mit einem Schreibschutz versehen, wenn Sie die Schreibschutzkennung setzen.

Anzeige des Speicherausbaus

Wenn Sie sich im EPROM-Betrieb mit der PG-Funktion "Speicherausbau" Informationen über den Speicher ausgeben lassen, so wird als Länge des Speichers der Wert '0' und als Endadresse des Anwenderspeichers der Wert '0EEEEh' angezeigt.

Hinweis!

Die Memory Card kann nur am PG programmiert werden. Dazu können Sie die PG-Software ab Version 6 verwenden.

Beim Programmieren müssen Sie am PG die Betriebsart "WORD" wählen.

DB-RAM

In das DB-RAM werden Datenbausteine DB/DX durch Erzeugen oder Kopieren geschrieben. Beim Übertragen von Datenbausteinen vom PG in die CPU werden diese im DB-RAM abgelegt, falls der Anwenderspeicher gefüllt ist oder im "RAM-Betrieb" "Alternatives Laden eingestellt ist.

Bearbeitung des Anwenderprogramms

Die gesamte Software auf der CPU (bestehend aus Systemprogramm und Anwenderprogramm) hat folgende Aufgaben zu bearbeiten:

- ANLAUF der CPU,
- Steuerung eines Automatisierungsprozesses durch ständig sich wiederholende Operationsfolgen (ZYKLUS).
- Steuerung eines Automatisierungsprozesses durch Reaktion auf Ereignisse, die sporadisch oder zu bestimmten Zeiten eintreffen (Alarme) sowie Reaktion auf Fehler.

Bei allen drei Aufgaben haben Sie die Möglichkeit, über Anwenderschnittstellen (Organisationsbausteine OB 1 bis OB 35) spezielle Teile Ihres Anwenderprogramms auf der CPU ablaufen zu lassen.

Anlauf

Bevor die CPU in die zyklische Bearbeitung eintreten kann, ist es u.U. erforderlich, eine Initialisierung durchzuführen, um einen definierten Ausgangszustand für die zyklische Bearbeitung herzustellen und z.B. Zeitraster für die Ausführung bestimmter Funktionen vorzugeben. Welche Initialisierung dies ist, hängt von dem Ereignis ab, das zu einem Anlauf führt, sowie von Einstellungen, die Sie an Ihrer CPU vornehmen.

Sie können das Verhalten der CPU bei einem Anlauf beeinflussen durch Programmieren der Organisationsbausteine OB 20, OB 21 und OB 22 oder durch Parametrieren des Datenbausteins DX 0.

Zyklus

Nachdem ein Anlauf durchgeführt worden ist, tritt das Systemprogramm in die zyklische Bearbeitung ein. Es übernimmt dabei Hintergrundfunktionen, die für Automatisierungsaufgaben erforderlich sind (siehe Abb. am Kapitelanfang).

Nach Ausführung der Systemfunktionen zu Beginn eines Zyklus wird vom Systemprogramm als zyklisches Anwenderprogramm der Organisationsbaustein OB 1 oder der Funktionsbaustein FB 0 aufgerufen. In einem dieser Bausteine programmieren Sie die Operationen für die zyklische Bearbeitung.

Reaktionen bei Alarmen und Fehlern

Um auf einen Alarm oder Fehler speziell reagieren zu können, stehen Ihnen auf der CPU 928 spezielle Organisationsbausteine (OB 2und OB 9 bis OB 18 für Alarmbearbeitung, OB 19 und OB 23 bis OB 35 für Reaktionen im Fehlerfall) zur Verfügung, in denen Sie das entsprechende Programm hinterlegen können.

Bei der Alarm- oder Fehlerbearbeitung wird vom Systemprogramm der zugehörige Organisationsbaustein in die zyklische Bearbeitung "eingeschachtelt". Dies bedeutet, dass die zyklische Bearbeitung durch die Bearbeitung eines Alarms oder Fehlers unterbrochen wird.

Die Einschachtelung der entsprechenden Organisationsbausteine erfolgt nach einem festen Prioritätenschema.

Außer durch die genannten Organisationsbausteine können Sie das Verhalten der CPU bei der Alarmbearbeitung und im Fehlerfall durch Parametrieren des Datenbausteins DX 0 beeinflussen.

Die Organisationsbausteine OB 1 bis OB 39 können vom Systemprogramm aufgerufen werden, sobald sie in den Programmspeicher geladen worden sind (auch im laufenden Betrieb).

Werden sie nicht geladen, so erfolgt entweder keine Reaktion der CPU oder - in den meisten Fehlerfällen - sie geht in den Stoppzustand.

Wie die Organisationsbausteine lässt sich auch der Datenbaustein DX 0 im laufenden Betrieb in den Programmspeicher laden. Er wird jedoch erst mit dem nächsten Neustart wirksam. Wird der DX 0 nicht geladen, gelten die Standardeinstellungen.

Begriffs-definitionen

Zykluszeit

Der Zyklus beginnt mit der Triggerung der sogenannten Zykluszeitüberwachung und endet bei der nächsten Triggerung. Die Zeit, die die CPU für die Programmbearbeitung zwischen zwei Triggerungen benötigt, wird Zykluszeit genannt. Sie setzt sich aus der Laufzeit des Systemprogramms und der Laufzeit des Anwenderprogramms zusammen.

In die Zykluszeit geht somit ein:

- die Bearbeitungszeit des zyklischen Programms (System- und Anwenderprogramm),
- die Bearbeitungszeit von Alarmen (z.B. zeitgesteuerter Alarm),
- die Bearbeitungszeit von Unterbrechungen (Fehlern).

Zykluszeitüberwachung

Die Zykluszeit wird von der CPU auf einen Maximalwert überwacht. Standardmäßig ist dieser Maximalwert auf 150 ms eingestellt. Als Anwender haben Sie die Möglichkeit, die Zykluszeitüberwachung selbst einzustellen bzw. sie während der Anwenderprogrammbearbeitung neu zu starten.

Prozessabbild der Ein- und Ausgänge (PAE und PAA)

Vor Beginn der zyklischen Anwenderprogrammbearbeitung liest das Systemprogramm die Signalzustände der Eingabe-Peripheriebaugruppen in das Prozessabbild der Eingänge ein. Das Anwenderprogramm wertet die Signalzustände im Prozessabbild der Eingänge aus und setzt in Abhängigkeit dieser Auswertung die Signalzustände für die Ausgänge im Prozessabbild der Ausgänge. Nach Bearbeitung des Anwenderprogramms überträgt das Systemprogramm die Signalzustände des Prozessabbilds der Ausgänge zu den Ausgabe-Peripheriebaugruppen.

Durch die Zwischenspeicherung der Peripheriesignale im Prozessabbild der Ein- und Ausgänge wird vermieden, dass ein Ändern des logischen Zustandes eines Bits innerhalb eines Programmzyklus zum "Flattern" des zugehörigen Peripherieausgangs führt.

Das Prozessabbild ist somit ein Speicherbereich, dessen Inhalt nur einmal pro Zyklus an die Peripherie ausgegeben bzw. von der Peripherie eingelesen wird.

Hinweis!

Ein Prozessabbild existiert nur für Ein- und Ausgabebytes der P-Peripherie mit Byteadressen von 0 bis 127.

Unterbrechungsereignisse

Die zyklische Programmbearbeitung kann unterbrochen werden durch

- eine prozessalarmgesteuerte Programmbearbeitung,
- eine zeitgesteuerte Programmbearbeitung,
- einen Verzögerungsalarm,
- einen uhrzeitgesteuerten Weckalarm.

Sie kann unterbrochen bzw. ganz abgebrochen werden

- beim Auftreten eines Geräte- oder Programmfehlers,
- durch Bedienung (PG-Funktion, Stoppschalter, MP-STP),
- durch eine Stoppoperation.

Operationen mit Beispielen

Eine Operation setzt sich zusammen aus einem Operationsteil und einem Operanden. Im Operationsteil wird festgelegt, was durch die CPU getan werden soll (Operation). Der Operandenteil gibt an, womit eine Operation durchgeführt werden soll.

Die Operationen lassen sich in verschiedene Gruppen einteilen:

- Grundoperationen
  (in allen Codebausteinen anwendbar),
- Ergänzende Operationen
  (nur in Funktionsbausteinen FB/FX anwendbar),
- Organisatorische Operationen
  (nur in Funktionsbausteinen FB/FX anwendbar),
- Semaphor-Operationen
  (nur in Funktionsbausteinen FB/FX anwendbar).

Akkumulatoren als Arbeitsregister

Die CPU 928 besitzt vier Akkumulatoren, AKKU 1 bis AKKU 4. Ein überwiegender Teil der Operationen verwendet als Quelle für die Operanden und als Ziel für die Ergebnisse zwei Register (32 bit): AKKU 1 und AKKU2.

AKKU 1[1)]

← High-Word →		← Low-Word →	
High-Byte	Low-Byte	High-Byte	Low-Byte
31 24	23 16	15 8	7 0
AKKU-1-H-H	AKKU-1-H-L	AKKU-1-L-H	AKKU-1-L-L
AKKU-1-H		AKKU-1-L	

[1)] die Aufteilung bei AKKU 2 bis AKKU 4 ist analog

Die Akkumulatoren werden abhängig von der auszuführenden Operation beeinflusst, z.B.:

- bei den Ladeoperationen wird als Ziel immer der AKKU 1 verwendet; der alte Inhalt von AKKU 1 wird in den AKKU 2 (Stack Lift) geschoben; die Akkumulatoren 3 und 4 werden bei allen Ladeoperationen nicht verändert;
- arithmetische Operationen verknüpfen den Inhalt von AKKU 1 und AKKU 2, schreiben das Ergebnis in den AKKU 1 und übertragen den Inhalt des AKKU 3 nach AKKU 2 und den Inhalt des AKKU 4 nach AKKU 3 (Stack Drop); bei 16-bit-Festpunktarithmetik werden nur das Low-Word von AKKU 3 in das Low-Word von AKKU 2 und das Low-Word von AKKU 4 in das Low-Word von AKKU 3 übertragen;
- beim Addieren einer Konstanten (ADD BF/KF/DH) zum Inhalt des AKKU 1 werden die Akkumulatoren 2, 3 und 4 nicht verändert.

Ergebnisanzeigen

Operationen sind entweder anzeigenbildend oder anzeigenabfragend. Die Anzeigen werden in einem Anzeigenbyte hinterlegt. Bei den Anzeigen sind zwei Gruppen zu unterscheiden: Anzeigen von digitalen Operationen (Wortanzeigen - Bit 4 bis 7 im Anzeigenbyte) und Anzeigen von binären und organisatorischen Operationen (Bit-Anzeigen - Bit 0 bis 3 im Anzeigenbyte). Wie die verschiedenen Anzeigen durch Operationen beeinflusst bzw. ausgewertet werden können, entnehmen Sie bitte der Operationsliste.

Das Anzeigenbyte wird bei der PG-Online-Funktion "Status Baustein" angezeigt und hat folgenden Aufbau:

Wort-Anzeigen				Bit-Anzeigen			
ANZ1	ANZ0	OV	OS	OR	STA	VKE	$\overline{\text{ERAB}}$
Bit 7	6	5	4	3	2	1	0

Bit-Anzeigen

$\overline{\text{ERAB}}$ Erstabfrage

Eine logische Verknüpfungskette mit binären Verknüpfungen beginnt immer mit einer sogenannten Erstabfrage, bei der das VKE neu gebildet wird. Mit Ihr wird die Bit-Anzeige $\overline{\text{ERAB}}$ = 1 gesetzt. Im Laufe weiterer logischer Verknüpfungen in der begonnenen Kette bleibt $\overline{\text{ERAB}}$ = 1, und das VKE kann durch diese logischen Verknüpfungen verändert werden.

Die begonnene Verknüpfungskette wird beendet durch eine binäre Speicheroperation (z.B. S A 5.0). Mit der Speicheroperation wird $\overline{\text{ERAB}}$ = 0 gesetzt; das VKE kann jetzt nur noch ausgewertet (z.B. durch VKE-abhängige Befehle), aber nicht mehr weiter verknüpft werden. Die nächste binäre logische Verknüpfung nach einer binären Speicheroperation ist wieder eine "Erstabfrage".

Beispiel zu $\overline{\text{ERAB}}$

:S	A	7.7	letzte Operation der vorhergehenden
:			Verknüpfungskette
:U	E	1.0	$\overline{\text{ERAB}}$ wird auf '1'gesetzt, VKE wird durch
:			UND-Verknüpfung neu gebildet
:O	E	6.3	VKE wird durch ODER-Verknüpfung
:			beeinflusst
:UN	E	2.1	VKE wird durch UND-NICHT-Verknüpfung
:			beeinflusst
:S	A	2.4	$\overline{\text{ERAB}}$ wird auf '0'gesetzt, Verknüpfungskette
:			ist beendet
SPB	FB	150	Funktionsbaustein in Abhängigkeit vom VKE
:			aufrufen

weitere Bit-Anzeigen

- VKE Verknüpfungsergebnis
  Ergebnis binärer Verknüpfungen, Wahrheitsaussage bei den Vergleichsbefehlen.
- STA Status
  Gibt bei Bit-Befehlen den logischen Zustand des gerade abgefragten oder gesetzten Bits an. Der Status wird bei binären Verknüpfungsoperationen - ausgenommen U(, O(,),O - und bei Speicheroperationen aktualisiert.
- OR Oder
  Interne Anzeige der CPU für die Behandlung von "UND-vor-ODER-Verknüpfungen".

Wortanzeigen

- OV Overflow
  Gibt an, ob bei der eben abgeschlossenen arithmetischen Operation der zulässige Zahlenbereich überschritten worden ist.
- OS Overflow speichernd

OS dient dazu, im Verlaufe mehrerer arithmetischer Operationen zu erkennen, ob irgendwann ein Überlauf (Overflow) aufgetreten ist.

- ANZ 1 und ANZ 0

Codierte Ergebnisanzeigen, deren Interpretation aus der Tabelle unten ersichtlich wird.

Zur unmittelbaren Auswertung der Anzeigen stehen Vergleichs- und Sprungoperationen zur Verfügung

Wortanzeigen		Arithmetische Operationen	Digitale Verknüpfungs-	Vergleichsoperationen	Schiebeoperationen	Bei SES, SEF	Ausgeführte Sprungoperation
ANZ1	ANZ0		operationen				
0	0	Ergebnis = 0	Ergebnis = 0	AKKU 2 = AKKU 1	geschobenes Bit = 0	Semaphor ist gesetzt	SPZ
0	1	Ergebnis < 0	-	AKKU 2 < AKKU 1	-	-	SPM SPN
1	0	Ergebnis > 0	Ergebnis ≠ 0	AKKU 2 > AKKU 1	geschobenes Bit = 1	Semaphor wird gesetzt bzw. freigegeben	SPP SPN
1	1	Division durch 0	-	-	-	-	SPN

Hinweis!

Bei einem Ebenenwechsel, z.B. bei der Behandlung eines Weckalarms, werden alle Akkumulatoren sowie die Bit- und Wortanzeigen (VKE usw.) gerettet und bei Fortsetzung der unterbrochenen Ebene wieder geladen

Grund-operationen

Die Grundoperationen können Sie in allen Codebausteinen und in allen Darstellungsarten (KOP, FUP und AWL) benutzen.

Binäre Verknüpfungs-operationen

Operation	Operand			Funktion
U				UND-Verknüpfung mit Abfrage auf Signalzustand "1"
O				ODER-Verknüpfung mit Abfrage auf Signalzustand "1"
	E	0.0 bis	127.7	eines Eingangs im PAE
	A	0.0 bis	127.7	eines Ausgangs im PAA
	M	0.0 bis	255.7	eines Merkerbits
	S	0.0 bis	1023.7	eines S-Merkerbits
	D	0.0 bis	215.15	eines Datenwortbits
	T	0 bis	255	einer Zeit
	Z	0 bis	255	eines Zählers
UN				UND-Verknüpfung mit Abfrage auf Signalzustand "0"
ON				ODER-Verknüpfung mit Abfrage auf Signalzustand "0"
	E	0.0 bis	127.7	eines Eingangs im PAE
	A	0.0 bis	127.7	eines Ausgangs im PAA
	M	0.0 bis	255.7	eines Merkerbits
	S	0.0 bis	1023.7	eines S-Merkerbits
	D	0.0 bis	215.15	eines Datenworts
	T	0 bis	255	einer Zeit
	Z	0 bis	255	eines Zählers
O	-			ODER-Verknüpfung von UND-Funktionen
U(O( )	-			UND-Verknüpfung von Klammerausdrücken ODER-Verknüpfungen von Klammerausdrücken Klammer zu (Abschluss eines Klammerausdrucks) Es sind maximal 8 Ebenen zulässig, d.h. 7 geöffnete Klammern

VKE-Bildung

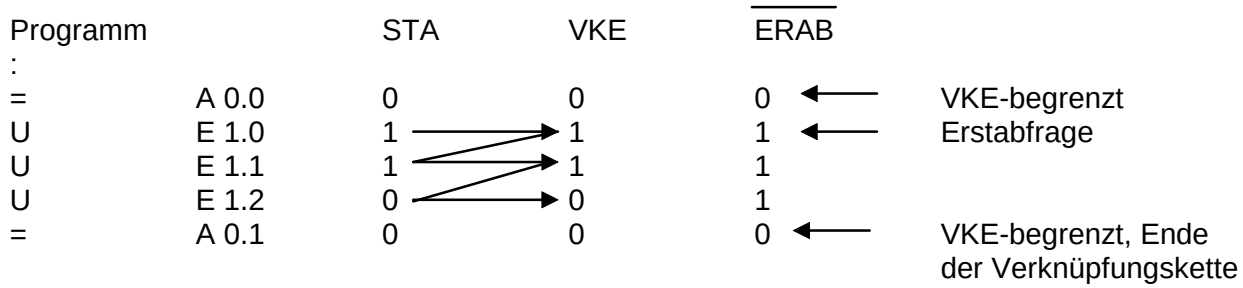
Die binären Verknüpfungsoperationen erzeugen das Verknüpfungsergebnis (VKE).

Am Anfang einer Verknüpfungskette hängt die Bildung des VKE (Erstabfrage) nur vom abgefragten Signalzustand (Status) ab, jedoch nicht von der Verknüpfungsart (O = ODER, U = UND).

Innerhalb einer Verknüpfungskette wird das VKE aus Verknüpfungsart, bisherigen VKE und dem abgefragten Signalzustand gebildet.

Eine Verknüpfungskette wird durch einen VKE-begrenzenden ($\overline{ERAB}$ = 0) Befehl (z.B. Speicheroperation) abgeschlossen. Danach kann das VKE zwar ausgewertet, jedoch nicht mehr weiter verknüpft werden.

Beispiel zur VKE-Bildung


Programm
STA
VKE
ERAB
:
= A 0.0 0 0 0 VKE-begrenzt
U E 1.0 1 1 1 Erstabfrage
U E 1.1 1 1 1
U E 1.2 0 0 1
= A 0.1 0 0 0 VKE-begrenzt, Ende der Verknüpfungskette

Speicheroperationen

Operation	Operand			Funktion
R				Bei Zustand '1' des VKE: Zurücksetzen
S				Bei Zustand '1'des VKE: Setzen
	E	0.0 bis	127.7	eines Eingangs im PAE
	A	0.0 bis	127.7	eines Ausgangs im PAA
	M	0.0 bis	255.7	eines Merkers
	S	0.0 bis	1023.7	eines S-Merkers
	D	0.0 bis	215.15	eines Bits im Datenwort
=				Zuweisung des VKE zu einem
	E	0.0 bis	127.7	Eingang im PAE
	A	0.0 bis	127.7	Ausgang im PAA
	M	0.0 bis	255.7	Merker
	S	0.0 bis	1023.7	S-Merker
	D	0.0 bis	215.15	Bit im Datenwort

Lade- und Transfer- operationen

Operation	Operand			Funktion
L T				Laden Transferieren
	EB	0 bis	127	eines Eingabebytes vom/zum PAE
	EW	0 bis	126	eines Eingabewortes vom/zum PAE
	ED	0 bis	124	eines Eingabe-Doppelwortes vom/zum PAE
	AB	0 bis	127	eines Ausgabebytes vom/zum PAA
	AW	0 bis	126	eines Ausgabewortes vom/zum PAA
	AD	0 bis	124	eines Ausgabe-Doppelwortes vom/zum PAA
	MB	0 bis	255	eines Merkerbytes
	MW	0 bis	254	eines Merkerwortes
	MD	0 bis	252	eines Merker-Doppelwortes
	SY	0 bis	1023	eines S-Merkerbytes
	SW	0 bis	1022	eines S-Merkerwortes
	SD	0 bis	1020	eines S-Merker-Doppelwortes
	DR	0 bis	255	des rechten Bytes eines DW aus bzw. nach DB,DX
	DL	0 bis	255	des linken Bytes eines DW aus bzw. nach DB,DX
	DW	0 bis	255	eines Datenwortes aus bzw. nach DB,DX
	DD	0 bis	254	eines Daten-Doppelwortes aus bzw. nach DB,DX
	PY	0 bis	255	eines Peripheriebytes der Analog- oder Digitaleingaben bzw. -ausgaben (P-Bereich)
	PW	0 bis	254	eines Peripheriewortes der Analog- oder Digitaleingaben bzw. -ausgaben (P-Bereich)
	QB	0 bis	255	eines Bytes der erweiterten Peripherie (Q-Bereich)
	QW	0 bis	254	eines Wortes der erweiterten Peripherie (Q-Bereich)

Fortsetzung...

...Fortsetzung

Operation	Operand		Funktion
L			Laden
	KB	0 bis 255	einer Konstanten, Bytewert
	KC	2 ASCII-Zeichen	einer Konstanten als 2 ASCII-Zeichen
	KF	-32768 bis +32767	einer Konstanten als Festpunktzahl
	KG	[1]	einer Konstanten als Gleitpunktzahl
	KH	0 bis FFFF	einer Konstanten als Hexadezimalzahl
	DH	0 - FFFF FFFF	einer Doppelwort-Konstanten als Hexadezimalzahl
	KM	16-bit-Muster	Seiner Konstanten als Bitmuster
	KY	0 bis 255 für jedes Byte	einer Konstanten als 2-Byte-Zahl
	KT	0.0 bis 999.3	einer Konstanten als Zeitwert (BCD-codiert)
	KZ	0 bis 999	einer Konstanten als Zählwert
	T	0 bis 255	eines Zeitwertes, dual-codiert
	Z	0 bis 255	eines Zählwertes, dual-codiert
LC			Laden eines
	T	0 bis 255	Zeitwertes
	Z	0 bis 255	Zählwertes im gepackten BCD-Code

[1] $\pm 0{,}1469368 \times 10^{-38}$ bis $\pm 0{,}1701412 \times 10^{39}$

Ladeoperationen

Ladeoperationen schreiben den adressierten Wert in den AKKU 1, dessen vorheriger Inhalt in den AKKU 2 gerettet wird (Stack Lift):

Transferoperationen

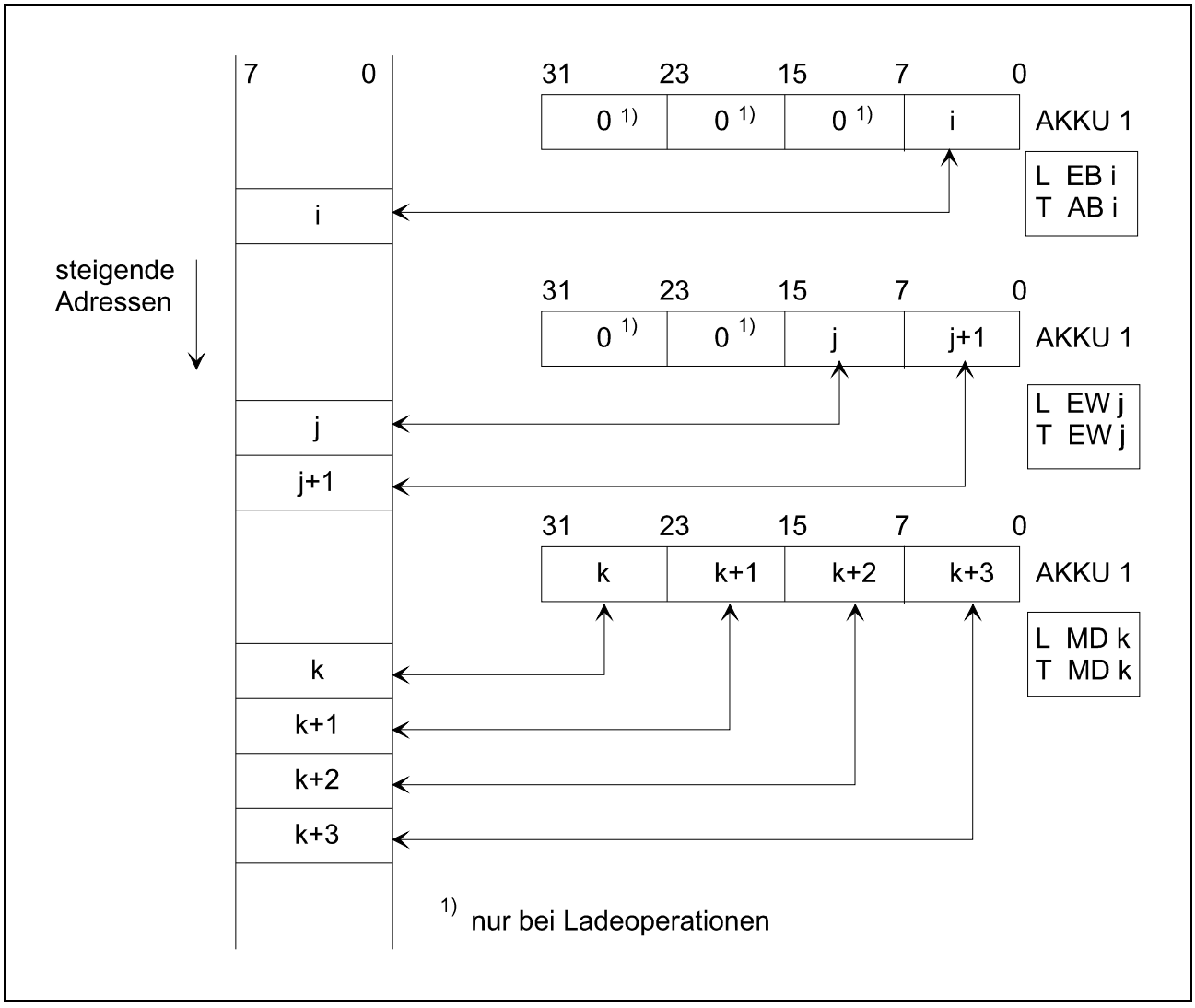
Transferoperationen schreiben den Inhalt des AKKU 1 in die adressierte Speicherzelle.

Beispiele Lade- und Transfer-operationen

Beispiel 1:

Das Bild zeigt das Laden/Transferieren eines Bytes, Wortes oder Doppelwortes aus einem/in einen byteweise organisierten Speicherbereich (PAE, PAA, Merker, Peripherie):

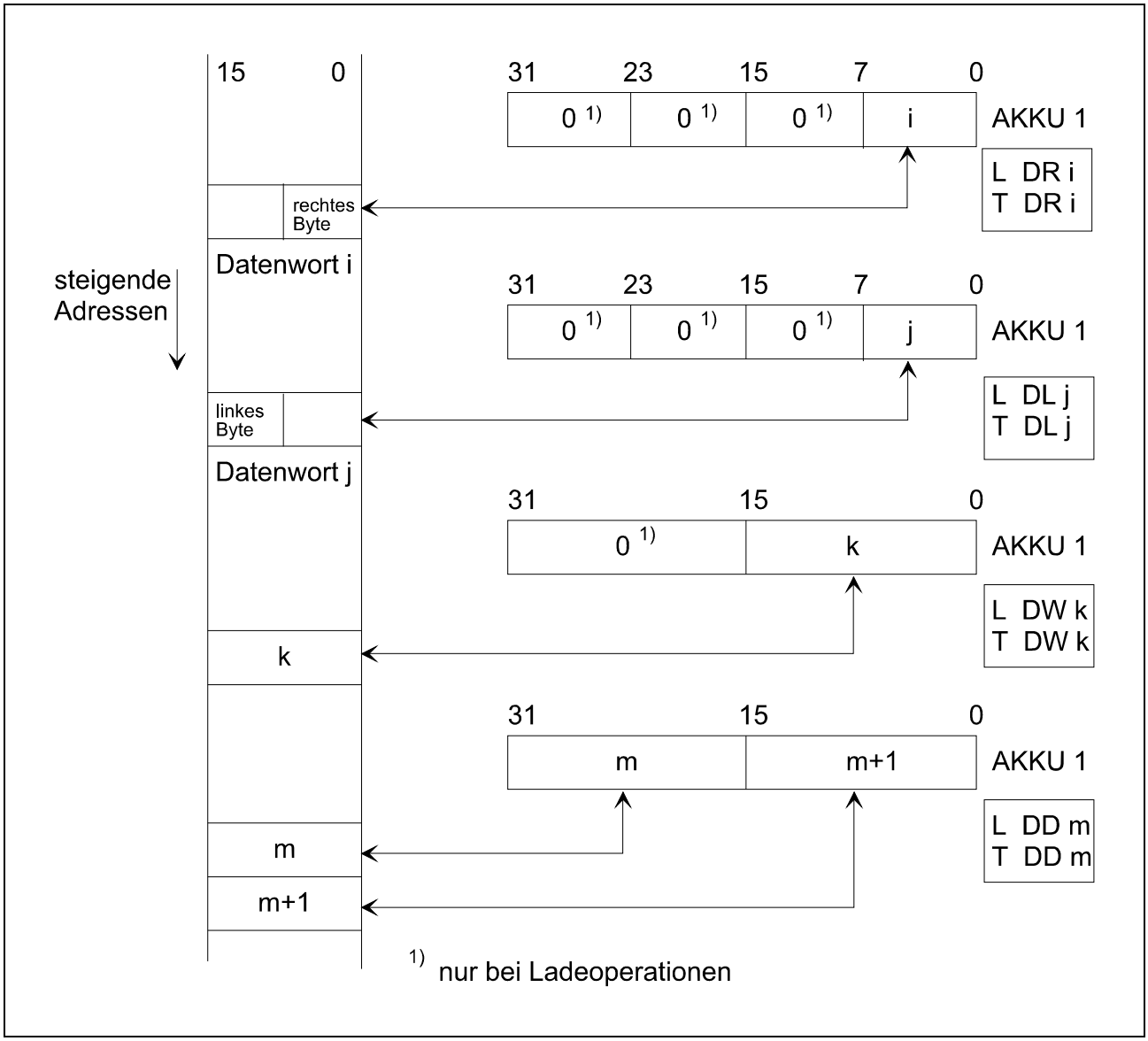
:L EB	i	geladen wird Byte i des PAE in den AKKU-1-LL
:L EW	j	geladen wird Byte j und j+1 des PAE in den AKKU-1-L
:L MD	k	geladen werden die Merkerbytes k bis k+3 in den AKKU 1



Beispiel 2:

Das Bild zeigt das Laden/Transferieren eines Bytes, Wortes oder Doppelwortes aus einem/in einen wortweise organisierten Speicherbereich.

:L DR	i	geladen wird das rechte Byte aus dem Datenwort i in den AKKU-1-LL
:L DL	j	geladen wird das linke Byte aus dem Datenwort j in den AKKU-1-LL
:L DW	k	geladen wird das Datenwort k in den AKKU-1-L
:L DD	m	geladen werden die Datenwörter m und m+1 in den AKKU 1



Hinweis!

Ladeoperationen beeinflussen die Anzeigen nicht.

Transferoperationen löschen das OS-Bit.

Beim Laden eines Bytes oder Wortes werden die höherwertigen Bytes im AKKU 1 gelöscht.

Ansprechen der Peripherie

Die Peripherie kann durch Lade- und Transferoperationen angesprochen werden:

- direkt:

  mit L../T....PY, ..PW, ..QB, ..QW

  oder
- über das Prozessabbild:

  mit L../T....EB, ..EW, ..ED, .AB, ..AW, ..AD

  und mit Verknüpfungsoperationen.

Hinweis!

Bei den Transferoperationen T PY 0 bis 127 und T PW 0 bis 126 wird parallel das Prozessabbild der Ausgänge nachgeführt.

Beachten Sie zur Peripherie folgende Punkte.

- Ein Prozessabbild der Ein- und Ausgänge existiert für je 128 Ein- und Ausgabebytes der P-Peripherie mit Byteadressen von 0 bis 127.
- Für den Bereich der P-Peripherie mit den relativen Byteadressen von 128 bis 255 und den gesamten Bereich der Q-Peripherie existiert kein Prozessabbild!
- Ein-/Ausgabebaugruppen mit Adressen der Q-Peripherie können nur in Erweiterungsgeräten stecken (nicht im Zentralgerät).
- In einem Erweiterungsgerät kann man entweder nur P-Peripherie oder nur Q-Peripherie verwenden.

Hinweis!

Falls in einem Erweiterungsgerät Relativadressen der Q-Peripherie verwendet werden, sind diese Adressen für Peripheriebaugruppen (P-Bereich) im Zentralgerät nicht mehr zulässig (Doppeladressierung!).

Zeit- und Zähloperationen

Um eine Zeit durch eine Startoperation oder einen Zähler durch eine Setzoperation zu laden, muss der Wert vorher in den AKKU 1 geladen werden.

Zu bevorzugen sind folgende Ladeoperationen:

Für Zeiten: L KT, L EW, L AW, L MW, L DW, L SW.

Für Zähler: L KZ, L EW, L AW, L MW, L DW, L SW.

Damit eine Zeit mit dem vorgegebenen Zeitwert gestartet wird, ist ein Flankenwechsel des VKE notwendig.

Ein Zähler wird mit dem vorgegebenen Zählwert gesetzt oder gezählt, wenn eine positive Flanke des VKE erkannt wird.

In der nachfolgenden Tabelle sind die Flankenwechsel in der Spalte VKE mit entsprechenden Pfeilen gekennzeichnet.

Operation	Operand		VKE [1]	Funktion
SI	T	0 bis 255	↑	Starten einer Zeit als Impuls
SV	T	0 bis 255	↑	Starten einer Zeit als verlängerter Impuls
SE	T	0 bis 255	↑	Starten einer Zeit als Einschaltverzögerung
SS	T	0 bis 255	↑	Starten einer Zeit als speichernde Einschaltverz.
SA	T	0 bis 255	↓	Starten einer Zeit als Ausschaltverzögerung
R	T	0 bis 255	1	Rücksetzen einer Zeit
S	Z	0 bis 255	↑	Setzen eines Zählers (BCD-Wert von 0 bis 999)
R	Z	0 bis 255	1	Rücksetzen eines Zählers
ZV	Z	0 bis 255	↑	Vorwärtszählen eines Zählers
ZR	Z	0 bis 255	↑	Rückwärtszählen eines Zählers

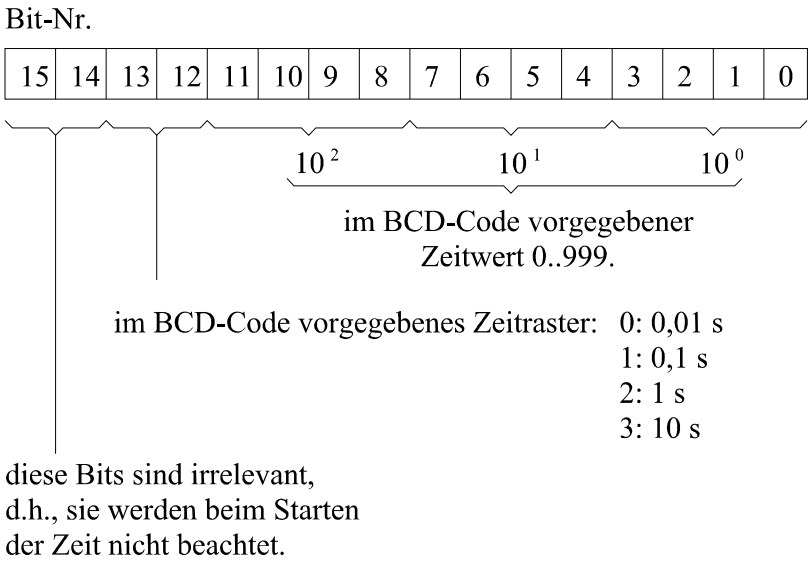
[1] positive Flanke (↑): Signalzustandsänderung von Zustand '0' nach Zustand '1'

negative Flanke (↓): Signalzustandsänderung von Zustand '1' nach Zustand '0'

Bei der Ausführung der Zeit- bzw. Zähloperation SI, SE, SV, SS, SA und S wird der im AKKU 1 stehende Wert in die Zeit- bzw. Zählstelle gebracht und die entsprechende Operation veranlasst.

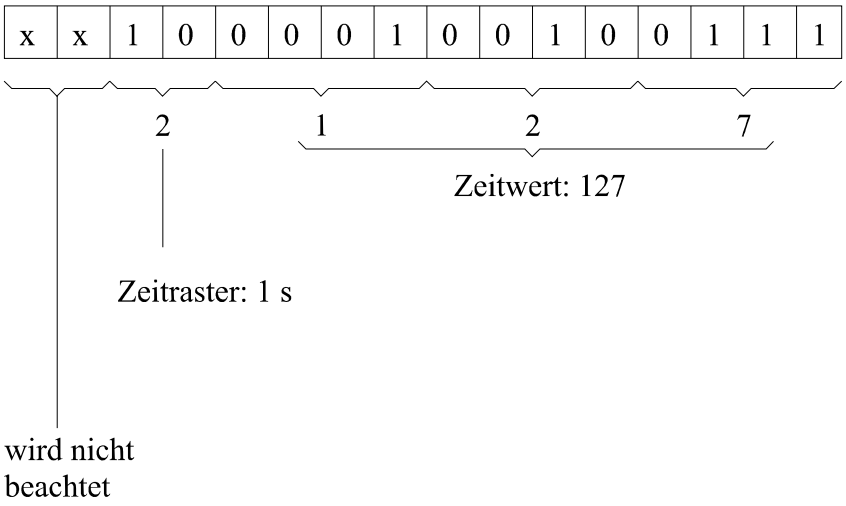
Zeitwert

Ein Zeitwert kann mit der Operation L KT direkt oder mit entsprechenden Ladeoperationen aus einem Merker- oder Datenwort indirekt in den AKKU 1 geladen werden. Er muss folgenden Aufbau haben. Das Zeitraster wird bei L KT hinter dem Punkt im Operanden angegeben:


Bit-Nr.
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
10 2 10 1 10 0
im BCD-Code vorgegebener Zeitwert 0..999.
im BCD-Code vorgegebenes Zeitraster: 0: 0,01 s
1: 0,1 s
2: 1 s
3: 10 s
diese Bits sind irrelevant, d.h., sie werden beim Starten der Zeit nicht beachtet.

Es soll eine Zeitdauer von 127 s vorgegeben werden:

Belegung der Bits:



Hinweis!

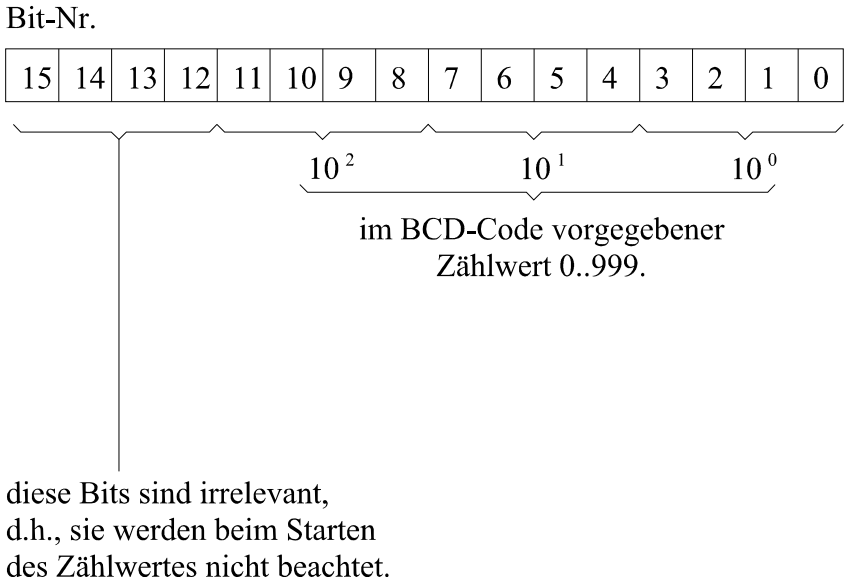
Jedes Starten einer Zeit ist mit einer Ungenauigkeit von 1 Zeitraster verbunden. Wählen Sie deshalb bei Verwendung von Zeitzellen ein möglichst kleines Zeitraster.

Beispiel:

Zeitdauer 4 s	nicht:	1 s x 4	Ungenauigkeit: 1 s
	sondern:	0,01 s x 400	Ungenauigkeit: 0,01 s

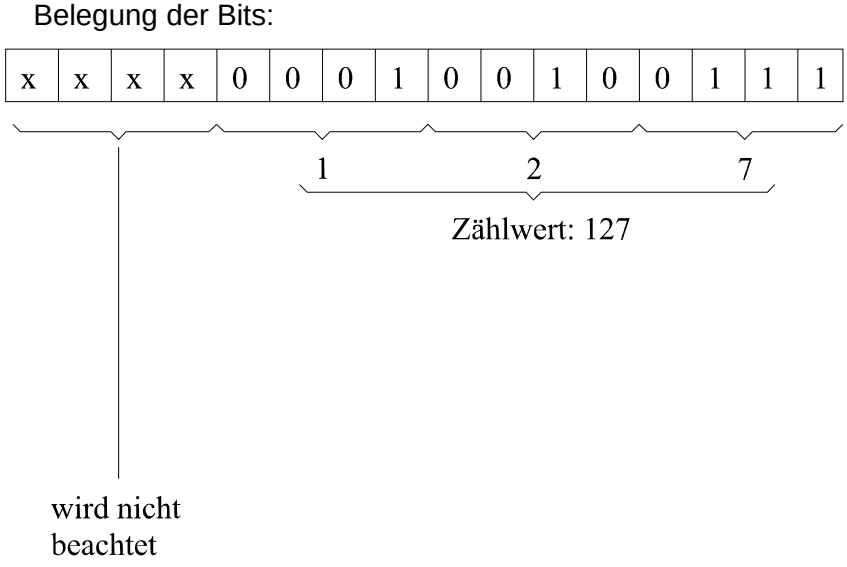
Zählwerte

Ein Zählwert kann mit dem Befehl L KZ direkt oder mit entsprechenden Ladeoperationen aus einem Merker- oder Datenwort indirekt in den AKKU 1 geladen werden. Er muss folgenden Aufbau haben:


Bit-Nr.
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
10 2
10 1
10 0
im BCD-Code vorgegebener
Zählwert 0..999.
diese Bits sind irrelevant,
d.h., sie werden beim Starten
des Zählwertes nicht beachtet.

Beispiel

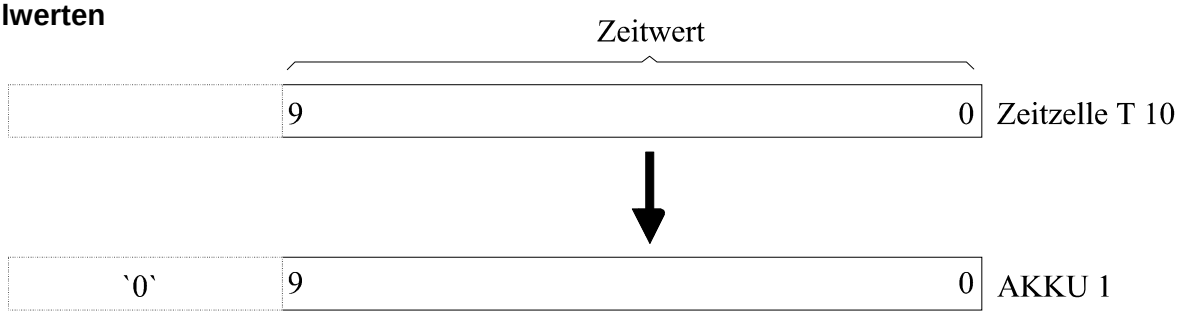
Es soll ein Zählwert von 127 vorgeben werden:



In der Zeit- bzw. Zählstelle selbst liegt der Zeit- bzw. Zählwert dualcodiert vor. Zur Abfrage der Zeit bzw. des Zählers kann der Wert der Zeit- bzw. Zählstelle direkt oder BCD-codiert in den AKKU 1 geladen werden.

Weitere Beispiele zu Zeit- und Zählwerten

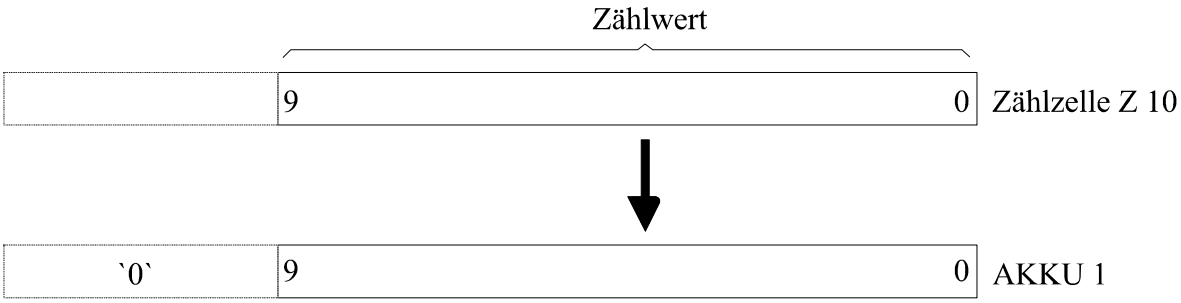
Direktes Laden von **Zeitwerten**:



"L T 10": Direktes Laden des dualen Zeitwertes der Zeit T 10 in den AKKU 1

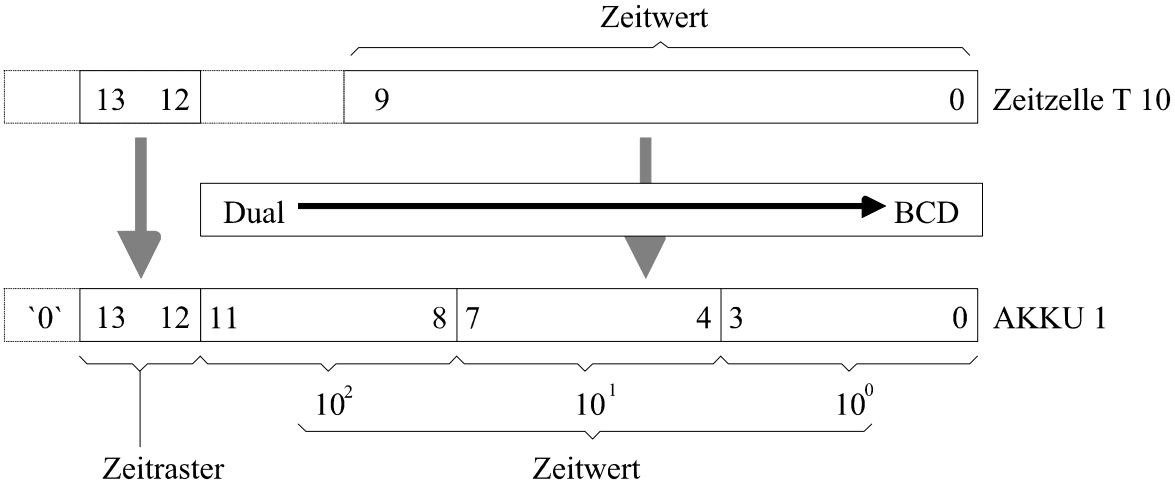
Das Zeitraster wird nicht mitgeladen.

Direktes Laden von **Zählwerten**:



"L Z 10": Direktes Laden des dualen Zählwertes des Zählers Z 10 in den AKKU 1

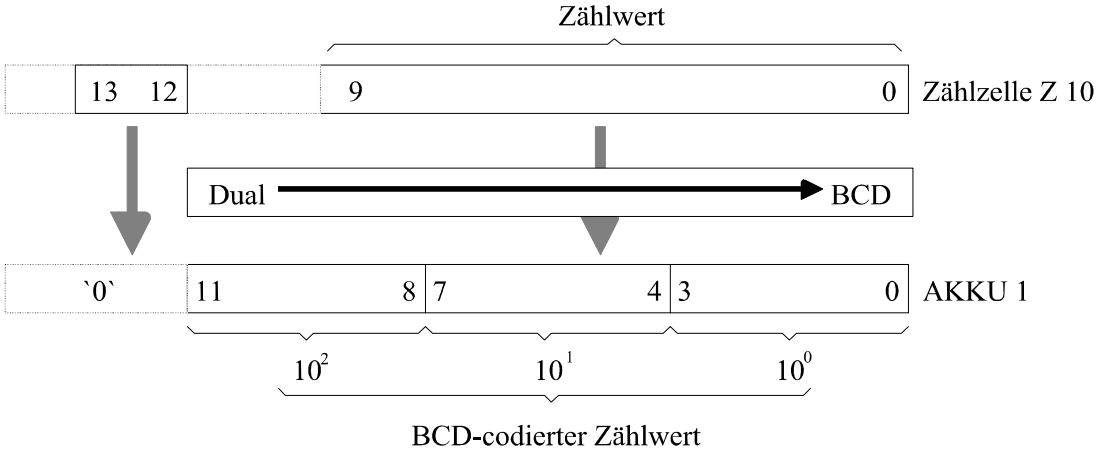
Codiertes Laden von **Zeitwerten**:



"LC T 10": Codiertes Laden des Zeitwertes und des Zeitrasters der Zeit T 10 in den AKKU 1

Das Zeitraster wird mitgeladen.

Codiertes Laden von **Zählwerten**:



"LC Z 10": Codiertes Laden des Zählwertes des Zählers Z 10 in den AKKU 1

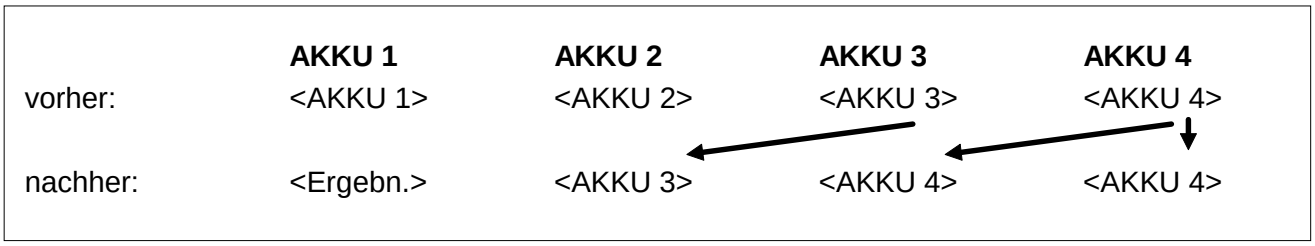
Beim codierten Laden werden die Zustandsbits 14 und 15 der Zeitzellen bzw. 12 bis 15 der Zählzellen nicht geladen. An ihrer Stelle steht 0 in AKKU 1. Der nun im Akku stehende Wert kann weiterverarbeitet werden.

Arithmetische Operationen

Die arithmetische Operationen verknüpfen den Inhalt von AKKU 1 und AKKU 2 (z.B. "AKKU 2-AKKU 1"). Das Ergebnis steht anschließend im AKKU 1.

Operation	Operand	Funktion
+ F	-	Addition zweier Festpunktzahlen (16 bit)
- F		Subtraktion zweier Festpunktzahlen (16 bit)
x F		Multiplikation zweier Festpunktzahlen (16 bit)
: F		Division zweier Festpunktzahlen (16 bit)
		Quotient in AKKU-1-L, Rest im AKKU-1-H
+ G		Addition zweier Gleitpunktzahlen (32 bit)
- G		Subtraktion zweier Gleitpunktzahlen (32 bit)
x G		Multiplikation zweier Gleitpunktzahlen (32 bit)
: G		Division zweier Gleitpunktzahlen (32 bit)

Die Rechenregister werden durch eine arithmetische Operation wie folgt verändert (bei Festpunkt-Operationen nur das Low-Word):



Hinweis!

Innerhalb der Ergänzenden Operationen stehen Ihnen Operationen zur Subtraktion und Addition von Doppelwort-Festpunktzahlen zur Verfügung.

Ferner können Sie von den Ergänzenden Operationen die Operation ENT für das Laden von AKKU 3 und AKKU 4 benutzen.

Vergleichs-operationen

Operation	Operand	Funktion
!=	-	Vergleich auf gleich
><	F	Vergleich auf ungleich
>	D	Vergleich auf größer
>=	G	Vergleich auf größer/gleich
<		Vergleich auf kleiner
<=		Vergleich auf kleiner/gleich
		...F: Vergleich zweier Festpunktzahlen (16 bit)
		...D: Vergleich zweier Festpunktzahlen (32 bit)
		...G: Vergleich zweier Gleitpunktzahlen (32 bit)

Baustein-operationen

Operation	Operand		Funktion
SPA			Sprung unbedingt
SPB			Sprung bedingt (nur wenn VKE = 1)
	OB 1	bis 39 [1)]	zu einem Organisationsbaustein
	OB 110	bis 255	zu einer Sonderfunktion
	PB 0	bis 255	zu einem Programmbaustein
	FB 0	bis 255	zu einem FB-Funktionsbaustein
	SB 0	bis 255	zu einem Schrittbaustein
BA			Sprung unbedingt
BAB			Sprung bedingt (nur wenn VKE = 1)
	FX 0	bis 255	zu einem FX-Funktionsbaustein
BE		-	Bausteinende
BEB			Bausteinende bedingt (nur wenn VKE = 1)
BEA			Bausteinende absolut
A	DB 3	bis 255	Aufschlagen eines DB-Datenbausteins
AX	DX 3	bis 255	Aufschlagen eines DX-Datenbausteins
E	DB 3	bis 255	Datenbaustein DB erzeugen
EX	DX 3	bis 255	Datenbaustein DX erzeugen
			(AKKU 1 muss die Anzahl der Datenwörter - maximal 4091- enthalten, die der neue Baustein haben soll.)

[1)] nur für Testzwecke

E DB/EX DX

Erzeuge Datenbaustein:

Die Operation E DB x erzeugt einen DB-Datenbaustein mit der Nummer x im Datenbaustein-RAM der CPU. Es gilt: ($3 \leq x \leq 255$)

Der Inhalt des Datenbausteins wird dabei nicht mit Null besetzt, d.h. die Datenwörter haben beliebige Inhalte.

Vor dem Programmieren der Anweisung müssen Sie die Anzahl der Datenwörter, die der neue DB haben soll, im AKKU-1-L hinterlegen. Der dazugehörige Bausteinkopf wird von der Operation "E DB" bzw. "EX DX" erzeugt. Ein so erzeugter Datenbaustein darf ohne Bausteinkopf maximal 4091 Wörter lang sein.

Falls der entsprechende Datenbaustein schon existiert, die Länge des DB unzulässig ist oder der Platz im DB-RAM nicht ausreicht, ruft das Systemprogramm den OB 31 auf. Wenn dieser nicht geladen ist, geht die CPU aufgrund eines Laufzeitfehlers in den Stoppzustand.

Die Operation EX DX erzeugt im DB-RAM einen DX-Datenbaustein und arbeitet wie E DB.

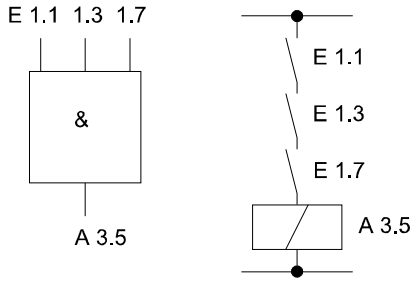
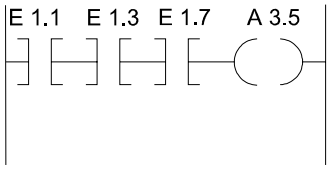
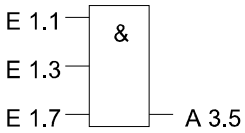
Null-/Bildaufbau-/Stopp-Operationen

Operation	Operand	Funktion
NOP0	-	Nulloperation
NOP1		Nulloperation
BLD	0 bis 255	Bildaufbau für das PG: wird von der CPU wie eine Nulloperation behandelt
STP	-	CPU geht in den STOP

Programmier-beispiele in AWL, KOP und FUP

Verknüpfungs-operationen

UND-Verknüpfung

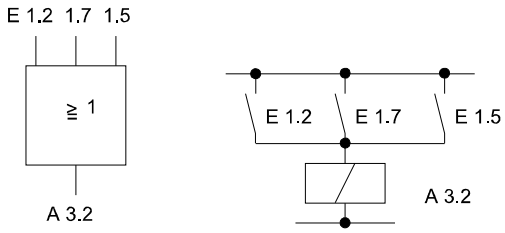
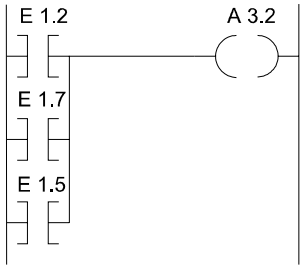
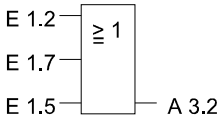
Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 1.1 U E 1.3 U E 1.7 = A 3.5		

Am Ausgang A 3.5 erscheint Signalzustand "1", wenn alle Eingänge gleichzeitig den Signalzustand "1" aufweisen.

Am Ausgang A 3.5 erscheint Signalzustand "0", wenn mindestens einer der Eingänge den Signalzustand "1" aufweist.

Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

ODER-Verknüpfung

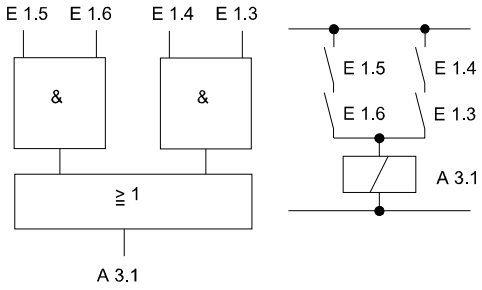
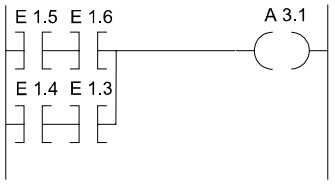
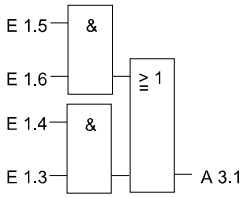
Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	O E 1.2 O E 1.7 O E 1.5 = A 3.2		

Am Ausgang A 3.2 erscheint Signalzustand "1", wenn mindestens einer der Eingänge den Signalzustand "1" aufweist.

Am Ausgang A 3.2 erscheint Signalzustand "0", wenn alle Eingänge gleichzeitig den Signalzustand "0" aufweisen.

Anzahl der Abfragen und die Reihenfolge der Programmierung ist beliebig.

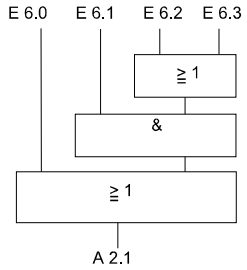
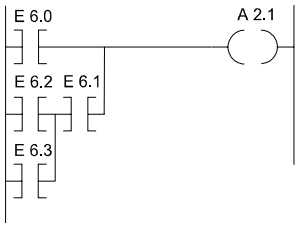
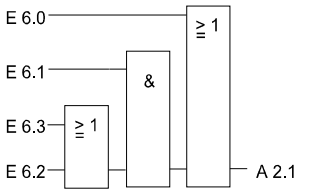
UND-vor-ODER-Verknüpfung

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 1.5 U E 1.6 O U E 1.4 U E 1.3 = A 3.1		

Am Ausgang A 3.1 erscheint Signalzustand "1", wenn mindestens eine UND-Verknüpfung erfüllt ist.

Am Ausgang A 3.1 erscheint Signalzustand "0", wenn keine UND-Verknüpfung erfüllt ist.

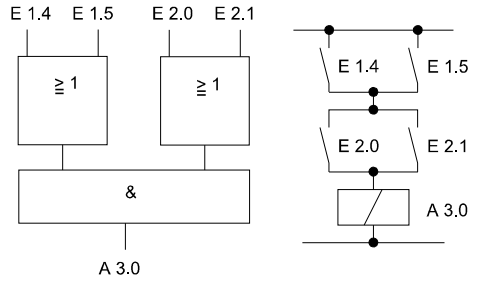
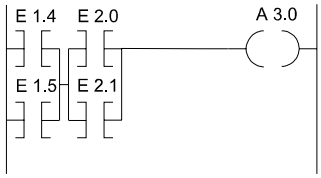
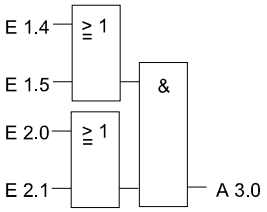
ODER-vor-UND-Verknüpfung *1. Beispiel*

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 6.0 O U E 6.1 U(O E 6.2 O E 6.3) = A 2.1		

Am Ausgang A 2.1 erscheint Signalzustand "1", wenn Eingang E 6.0 oder Eingang E 6.1 und einer der Eingänge E 6.2 bzw. E 6.3 Signal "1" führen.

Am Ausgang A 2.1 erscheint Signalzustand "0", wenn Eingang E 6.0 Signal "0" führt und die UND-Verknüpfung nicht erfüllt ist.

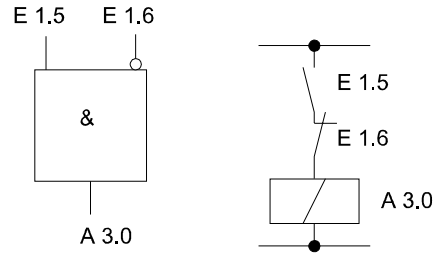
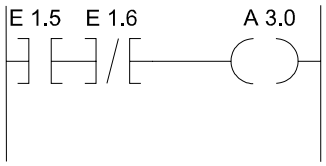
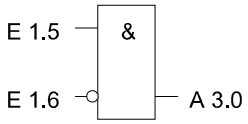
ODER-vor-UND-Verknüpfung *2.Beispiel*

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U(O E 1.4 O E 1.5) U(O E 2.0 O E 2.1) = A 3.0		

Am Ausgang A 3.0 erscheint Signalzustand "1", wenn beide ODER-Verknüpfungen erfüllt sind.

Am Ausgang A 3.0 erscheint Signalzustand "0", wenn mindestens eine ODER-Verknüpfung nicht erfüllt ist.

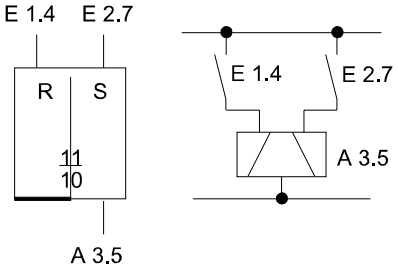
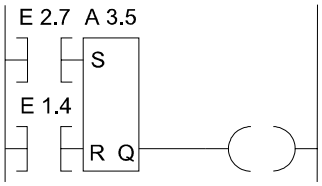
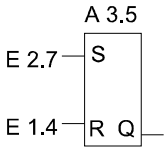
Abfrage auf Signalzustand "0"

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 1.5 UN E 1.6 = A 3.0		

Am Ausgang A 3.0 erscheint Signalzustand "1" nur dann, wenn Eingang E 1.5 den Signalzustand "1" (Schließer betätigt) und der Eingang E 1.6 den Signalzustand "0" (Öffner nicht betätigt) führt.

Speicher-operationen

RS-Speicherglied für speichernde Signalausgabe

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 2.7 S A 3.5 U E 1.4 R A 3.5		

Signalzustand "1" am Eingang E 2.7 bewirkt das Setzen des Speichergliedes (Signalzustand "1" am Ausgang A 3.5).

Wechselt der Signalzustand am Eingang E 2.7 nach "0", so bleibt dieser Zustand erhalten, d.h., das Signal wird gespeichert.

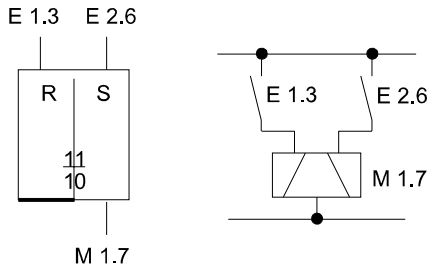
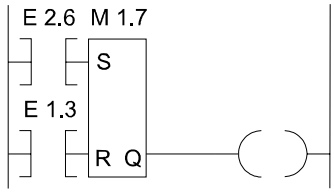
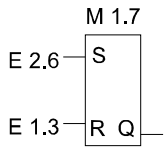
Signalzustand "1" am Eingang E 1.4 bewirkt das Rücksetzen des Speicher-gliedes (Signalzustand "0" am Ausgang A 3.5).

Wechselt der Signalzustand am Eingang E 1.4 nach "0", so bleibt dieser Zustand erhalten.

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 2.7) und des Rücksetzsignals (Eingang E 1.4) ist die zuletzt programmierte Abfrage (hier U E 1.4) während der Bearbeitung des übrigen Programms wirksam.

Das Rücksetzen ist vorrangig.

RS-Speicherglied mit Merkern

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 2.6 S M 1.7 U E 1.3 R M 1.7		

Signalzustand "1" am Eingang E 2.6 bewirkt das Setzen des Speichergliedes.

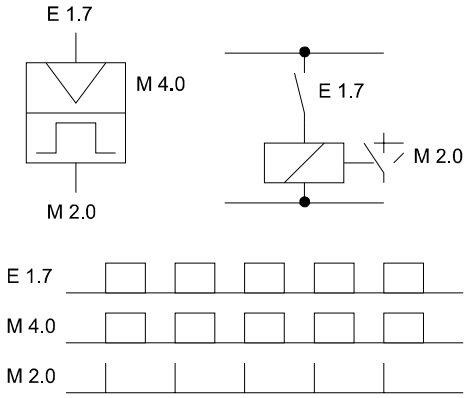
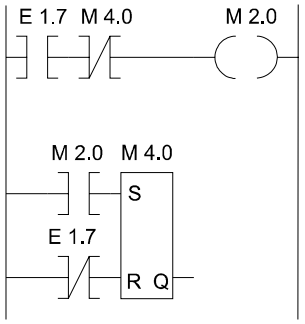
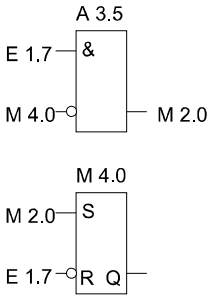
Wechselt der Signalzustand am Eingang E 2.6 nach "0", so bleibt dieser Zustand erhalten, d.h. das Signal wird gespeichert.

Signalzustand "1" am Eingang E 1.3 bewirkt das Rücksetzen des Speichergliedes.

Wechselt der Signalzustand am Eingang E 1.3 nach "0", so bleibt dieser Zustand erhalten.

Bei gleichzeitigem Anliegen des Setzsignals (Eingang E 2.6) und des Rücksetzsignals (Eingang E 1.3) ist die zuletzt programmierte Abfrage (hier U E 1.3) während der Bearbeitung des übrigen Programms wirksam Das Rücksetzen ist vorrangig.

Nachbildung eines Wischrelais

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 1.7 UN M 4.0 = M 2.0 U M 2.0 S M 4.0 UN E 1.7 R M 4.0		

Bei jeder ansteigenden Flanke des Eingangs E 1.7 ist die UND-Verknüpfung (U E 1.7 und UN M 4.0) erfüllt und mit VKE = "1" werden die Merker M 4.0 ("Flankenmerker") und M 2.0 ("Impulsmerker") gesetzt.

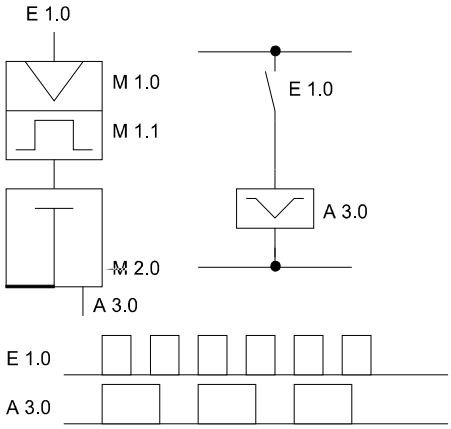
Beim nächsten Bearbeitungszyklus ist die UND-Verknüpfung U E 1.7 und UN M 4.0 nicht erfüllt, da der Merker M 4.0 gesetzt worden ist.

Der Merker M 2.0 wird rückgesetzt.

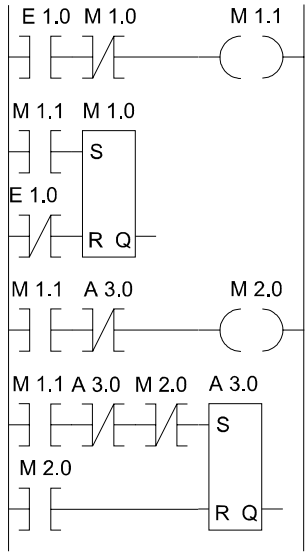
Der Merker M 2.0 führt also während eines einzigen Programmdurchlaufs Signalzustand "1".

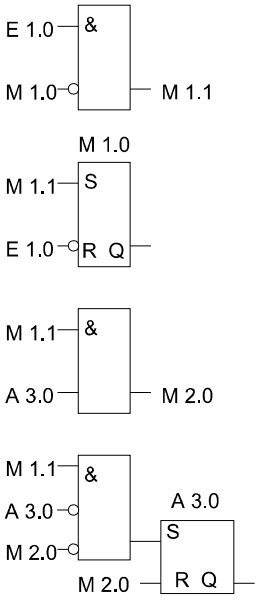
Binärumsetzer (T-Kippglied)

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan




```
U   E 1.0
UN  M 1.0
=   M 1.1
U   M 1.1
S   M 1.0
UN  E 1.0
R   M 1.0
U   M 1.1
U   A 3.0
=   M 2.0
U   M 1.1
UN  A 3.0
UN  M 2.0
S   A 3.0
U   M 2.0
R   A 3.0
```

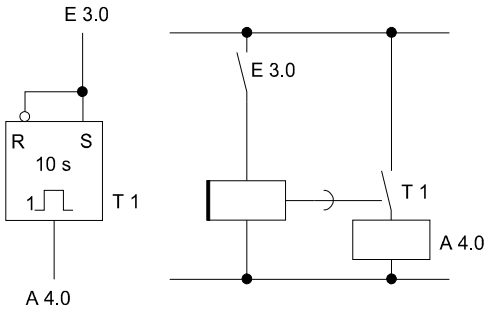
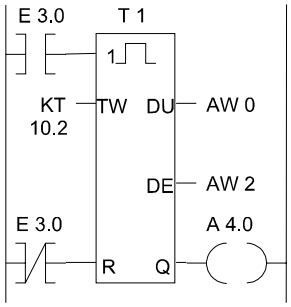
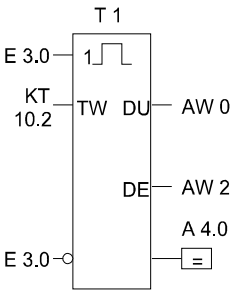




Der Binärumsetzer (Ausgang A 3.0) wechselt bei jedem Signalzustandswechsel von "0" nach "1" (ansteigende Flanke) des Eingangs E 1.0 seinen Zustand. Am Ausgang des Speichergliedes erscheint deshalb die halbe Eingangsfrequenz.

Zeitoperationen

Impuls

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 3.0 L KT 10.2 SI T 1 UN E 3.0 R T 1 L T 1 T AW 0 LC T 1 T AW 2 U T 1 = A 4.0		

Bei Signalzustandswechsel von "0" nach "1" am Eingang E 3.0 wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.

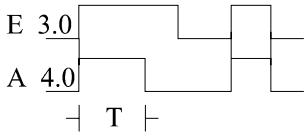
Bei Signalzustand "0" am Eingang E 3.0 wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand "1", solange die Zeit läuft.

KT 10.2

Das Zeitglied wird mit dem angegebenen Wert (10) geladen.

Die Zahl rechts vom Punkt gibt das Zeitraster an:

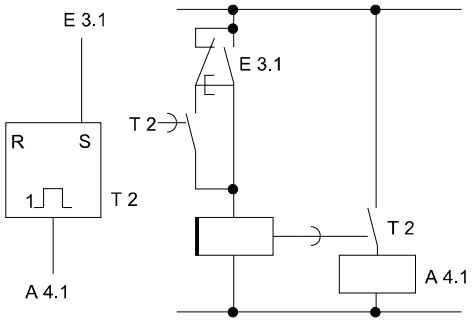
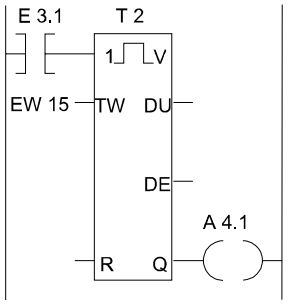
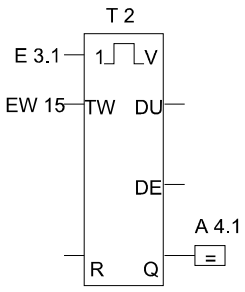


0 = 001 s 2 = 1 s

1 = 0.1 s 3 = 10 s

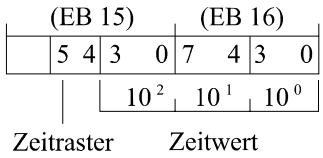
DU und DE sind digitale Ausgänge der Zeitzelle. Am Ausgang DU steht der Zeitwert dualcodiert, am Ausgang DE BCD-codiert mit Zeitraster an.

Verlängerter Impuls

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 3.1 L EW 15 SV T 2 U T 2 = A 4.1		

Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet.

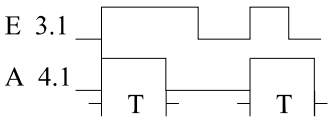
Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.
Die Abfragen U T bzw. O T liefern Signalzustand "1",



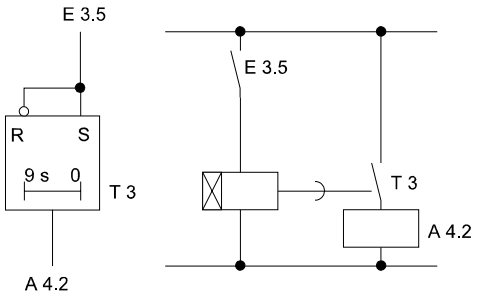
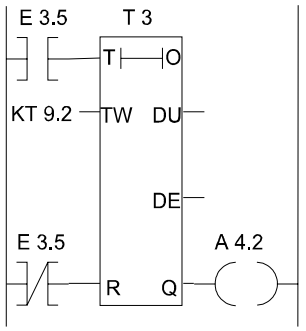
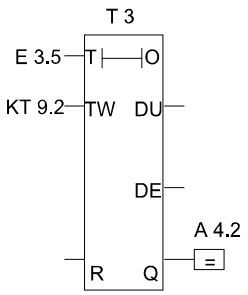
solange die Zeit läuft.

EW 15:

Setzen des Zeitwerts mit dem im BCD-Code vorliegenden Wert der Operanden E, A, M oder D (im Beispiel Eingangswort 15).



Einschaltverzögerung

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 3.5 L KT 9.2 SE T 3 UN E 3.5 R T 3 U T 3 = A 4.2		

Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied gestartet. Bei wiederholter Bearbeitung mit Verknüpfungsergebnis "1" bleibt das Zeitglied unbeeinflusst.

Bei Signalzustand "0" am Eingang E 3.5 wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand "1", wenn die Zeit abgelaufen ist und der Signalzustand "1" am Eingang E 3.5 noch ansteht.

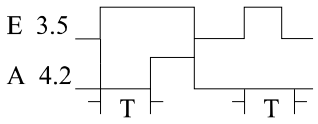
KT 9.2 :

Das Zeitglied wird mit dem angegeben Wert (9) geladen.

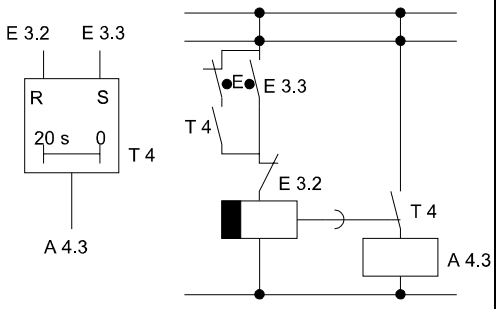
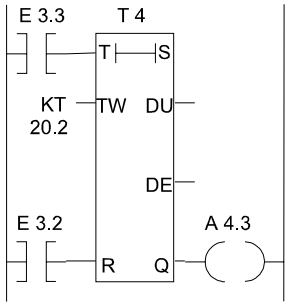
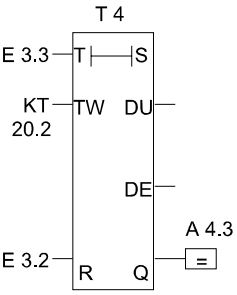
Die Zahl rechts vom Punkt gibt das Zeitraster an.

0 = 0.01 s 2 = 1 s

1 = 0.1 s 3 = 10 s

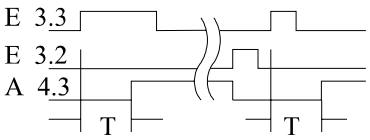


Speichernde Einschaltverzögerung

<table>
<tr><td rowspan="2">Aufgabenstellung</td><td colspan="3">Darstellung</td></tr>
<tr><td>Anweisungs
-liste</td><td>Kontaktplan</td><td>Funktionsplan</td></tr>
<tr>
<td> 
</td>
<td>U E 3.3
L KT20.2
SS T 4
U E 3.2
R T 4
U T 4
= A 4.3</td>
<td> 
</td>
<td> 
</td>
</tr>
</table>

Bei Verknüpfungsergebnis "1" und erstmaliger Bearbeitung wird das Zeitglied bearbeitet.

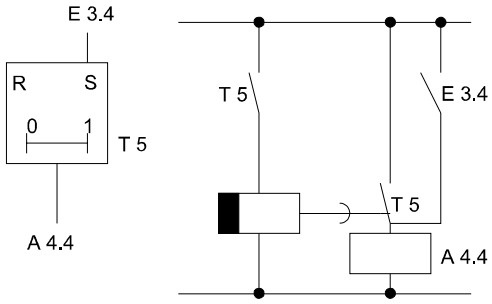
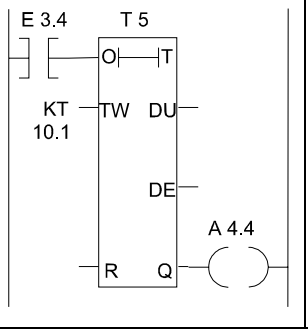
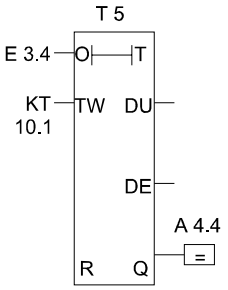
Bei Verknüpfungsergebnis "0" bleibt das Zeitglied unbeeinflusst.



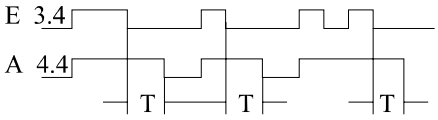
Die Abfragen U T bzw. O T liefern Signalzustand "1", wenn die Zeit abgelaufen ist.

Der Signalzustand wird erst dann "0", wenn das Zeitglied mit der Funktion RT zurückgesetzt worden ist.

Ausschaltverzögerung

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 3.4 L KT 10.1 SA T 5 U T 5 = A 4.4		

Wenn das Verknüpfungsergebnis am Starteingang von "1" nach "0" wechselt, wird die Zeit gestartet. Sie läuft mit der programmierten Zeitdauer ab.

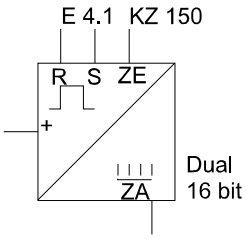
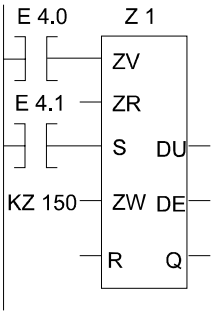
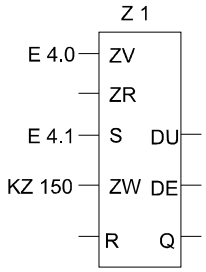


Bei Verknüpfungsergebnis "1" wird das Zeitglied auf Null gesetzt (gelöscht).

Die Abfragen U T bzw. O T liefern Signalzustand "1", wenn die Zeit läuft oder das Verknüpfungsergebnis am Eingang "1" ist.

Zähloperationen

Zähler setzen

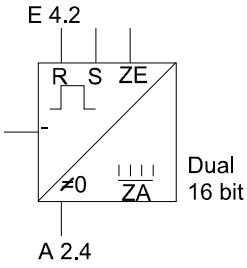
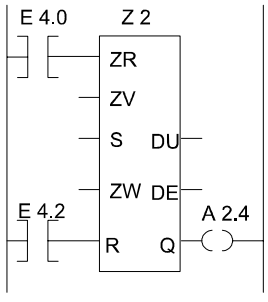
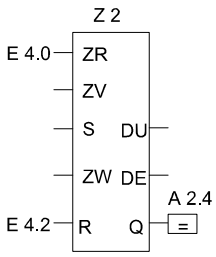
Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	U E 4.0 ZV Z 1 U E 4.1 L KZ 150 S Z 1		

Wenn das Verknüpfungsergebnis am Starteingang (E 4.1) von "0" nach "1" wechselt, wird der Zähler mit dem angegebenen Wert (150) geladen.

Der für die Flankenauswertung des Setzeingangs erforderliche Merker ist im Zählwort mitgeführt.

DU und DE sind digitale Ausgänge der Zählerzelle. Am Ausgang DU steht der Zählwert dualcodiert, am Ausgang DE BCD-codiert an.

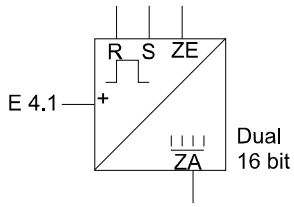
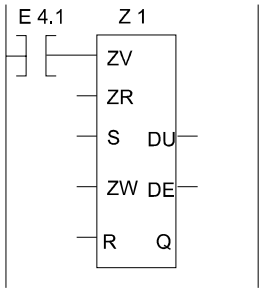
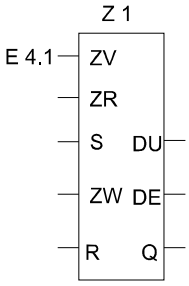
Zähler rücksetzen

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 4.0 ZR Z 2 U E 4.2 R Z 2 U Z 2 = A 2.4		

Bei Verknüpfungsergebnis "1" (E 4.2) wird der Zähler auf Null gesetzt (rückgesetzt).

Bei Verknüpfungsergebnis "0" bleibt der Zähler unbeeinflusst.

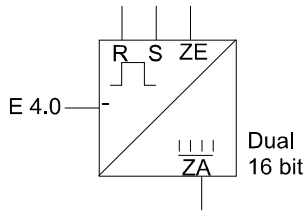
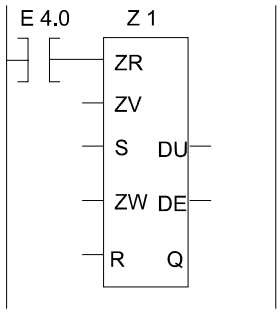
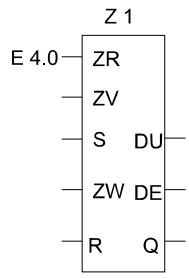
Vorwärts zählen

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 4.1 ZV Z 1		

Der Wert des adressierten Zählers wird um 1 erhöht, maximal bis zum Zählwert 999. Die Funktion ZV wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZV programmierten Verknüpfung ausgeführt. Die für die Flankenauswertung der Zähleingänge erforderlichen Merker sind im Zählwort mitgeführt.

Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.

Rückwärts zählen

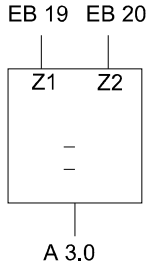
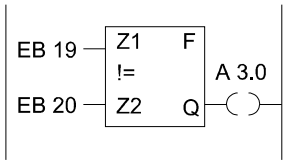
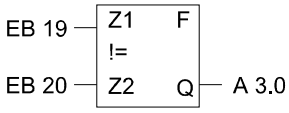
Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	U E 4.0 ZR Z 1		

Der Wert des adressierten Zählers wird um 1 erniedrigt, minimal bis zum Zählwert 0. Die Funktion wird nur bei einer positiven Flanke (von "0" nach "1") der vor ZR programmierten Verknüpfung ausgeführt. Die für die Flankenauswertung der Zähleingänge erforderlichen Merker sind im Zählwort mitgeführt.

Durch die zwei getrennten Flankenmerker für ZV und ZR kann ein Zähler mit zwei verschiedenen Eingängen als Vorwärts-/Rückwärtszähler verwendet werden.

Vergleichs-operationen

Vergleich auf gleich

Aufgabenstellung	Darstellung		
	Anweisungs-liste	Kontaktplan	Funktionsplan
	L EB 19 L EB 20 ! = F = A 3.0		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen.

Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = "1": Vergleich ist erfüllt, wenn AKKU-1-L gleich AKKU-2-L

VKE = "0": Vergleich ist nicht erfüllt, wenn AKKU-1-L ungleich AKKU-2-L

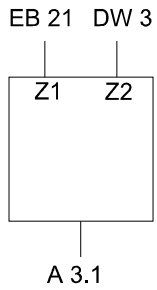
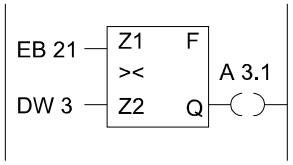
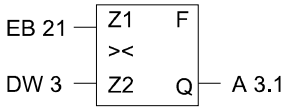
Die Anzeigen ANZ 1 und ANZ 0 werden, wie in der Operationsliste beschrieben, gesetzt.

AKKU-2-H und AKKU-1-H bleiben beim 16-bit-Festpunktvergleich an der Operation unbeteiligt.

Beim 32-bit-Festpunktvergleich (!=D) und Gleitpunktvergleich (!=G) werden die gesamten Inhalte von AKKU 1 und AKKU 2 (32 bit) miteinander verglichen.

Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d.h., der Inhalt von AKKU-1-L und AKKU-2-L wird hier als Festpunktzahl interpretiert.

Vergleich auf ungleich

Aufgabenstellung	Darstellung		
	Anweisungs -liste	Kontaktplan	Funktionsplan
	L EB 21 L DW 3 > < F = A 3.1		

Der zuerst angegebene Operand wird mit dem nachfolgenden Operanden entsprechend der Vergleichsfunktion verglichen.

Der Vergleich ergibt ein binäres Verknüpfungsergebnis.

VKE = "1": Vergleich ist erfüllt, wenn AKKU-1-L ungleich AKKU-2-L

VKE = "0": Vergleich ist nicht erfüllt, wenn AKKU-1-L gleich AKKU-2-L

Die Anzeigen ANZ 1 und ANZ 0 werden, wie in der Operationsliste beschrieben, gesetzt.

AKKU-2-H und AKKU-1-H bleiben beim 16-bit-Festpunktvergleich an der Operation unbeteiligt.

Beim 32-bit-Festpunktvergleich und Gleitpunktvergleich sind auch AKKU-2-H und AKKU-1-H beim Vergleich beteiligt. Entsprechendes gilt für die Vergleiche auf größer, größer gleich und kleiner gleich.

Beim Vergleich wird die Zahlendarstellung der Operanden berücksichtigt, d.h., der Inhalt von AKKU-1-L und AKKU-2-L wird hier als Festpunktzahl interpretiert.

Ergänzende Operationen

Den ergänzenden Operationsvorrat können Sie nur für Funktionsbausteine (FB und FX) verwenden. Der Gesamtoperationsvorrat für Funktionsbausteine besteht aus den Grundoperationen und den ergänzenden Operationen.

Zu den ergänzenden Funktionen gehören auch Systemoperationen: Mit den Systemoperationen können Sie z.B. den Speicher an beliebiger Stelle überschreiben oder den Inhalt der Arbeitsregister der CPU verändern.

Beachten Sie zum Thema "Systemoperationen" das Kapitel "Speicherzugriffe über absolute Adressen".

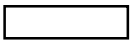
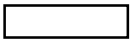
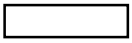
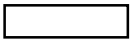
Hinweis!

Systemoperationen sollten nur von erfahrenen Programmierern und Systemkennern und nur mit Vorsicht angewendet werden

Bei den Funktionsbausteinen werden die Operationen nur in AWL dargestellt. Die Programme der Funktionsbausteine können also nicht in grafischer Form (KOP oder FUP) programmiert werden.

Im folgenden werden die ergänzenden Operationen beschrieben. Zusätzlich sind die Kombinationsmöglichkeiten der Substitutionsbefehle mit den Aktualoperanden angegeben.

Binäre Verknüpfungen

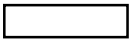
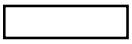
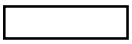
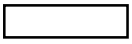
Operation			Operand	Funktion
	U	=		UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand '1'
	UN	=		UND-Funktion, Abfrage eines Formaloperanden auf Signalzustand '0'
	O	=		ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand '1'
	ON	=		ODER-Funktion, Abfrage eines Formaloperanden auf Signalzustand '0'
				Formaloperand einsetzen Als Aktualoperanden sind binär adressierte Eingänge, Daten und Merker (Parameterart: E, A; Parametertyp: BI) sowie Zeiten und Zähler (Parametertyp: T, Z) zugelassen.

Digital-verknüpfungen

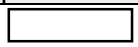
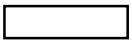
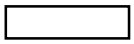
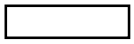
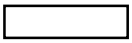
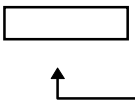
Operation		Operand	Funktion
	UW		UND-Verknüpfung von AKKU-1-L und AKKU-2-L
	OW		ODER-Verknüpfung von AKKU-1-L und AKKU-2-L
	XOW		Exklusiv-Oder-Verknüpfung von AKKU-1-L und AKKU-2-L

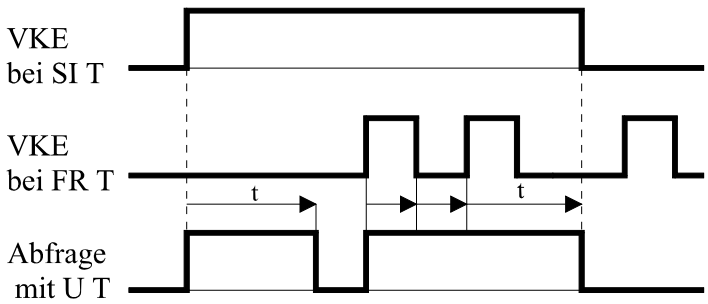
Die AKKUs 2,3 und 4 werden nicht beeinflusst, jedoch die Anzeigen ANZ 1 und ANZ 0.

Speicher-operationen

Operation		Operand	Funktion
	S =		Setzen (binär) eines Formaloperanden
	RB =		Rücksetzen (binär) eines Formaloperanden
	RD =		Rücksetzen (digital) eines Formaloperanden für Zeiten und Zähler
	= =		Zuweisen des Verknüpfungsergebnisses an einen Formaloperanden
			Formaloperand einsetzen Als Aktualoperanden sind binär adressierte Eingänge, Ausgänge und M-Merker zugelassen (Parameterart: E, A; Parametertyp: BI):

Zeit- und Zähloperationen

Operation		Operand	Funktion
	SI =		Eine Zeit (Formaloperand) als Impuls starten. Der Zeitwert muss in AKKU-1-L hinterlegt sein (Parameterart T).
	SE =		Eine Zeit (Formaloperand) einschaltverzögernd starten. Der Zeitwert muss in AKKU-1-L hinterlegt sein (Parameterart T).
	SVZ =		Eine Zeit (Formaloperand) als verlängerten Impuls starten mit dem im AKKU-1-L hinterlegten Zeitwert/einen Zähler (Formaloperand) setzen mit dem im AKKU-1-L hinterlegten Zählwert (Parameterart: T, Z).
	SSV =		Eine Zeit (Formaloperand) als speichernde Einschaltverzögerung starten mit dem im AKKU-1-L hinterlegten Wert bzw. Vorwärtszählen eines Zählers (Formaloperand); (Parameterart: T, Z)
	SAR =		Eine Zeit (Formaloperand) als Ausschaltverzögerung starten mit dem im AKKU-1-L hinterlegten Wert bzw. Rückwärtszählen eines Zählers (Formaloperand); (Parameterart: T, Z).
	FR =		Formaloperanden (Zeit/Zähler) für den Neustart freigeben (siehe "FR T, "FR Z"); (Parameterart: T, Z). Formaloperand einsetzen
	FR	T 0 bis 255 Z 0 bis 255	Eine Zeit für den Neustart freigeben: Die Operation wird nur bei steigender Flanke des VKE ausgeführt (Wechsel von '0' auf '1'). Die Zeit wird neu gestartet, wenn bei der Startoperation das VKE = '1' ist (siehe Skizze mit Zeitdiagramm unter der Tabelle). Einen Zähler für das Setzen bzw. Zählen freigeben: Die Operation wird nur bei steigender Flanke des VKE ausgeführt (Wechsel von '0' auf '1'). Der Zähler wird neu bearbeitet, wenn bei der Zähloperation das VKE = '1' ist.



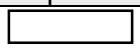
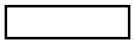
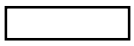
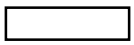
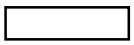
Beispiele

Funktionsaufruf			Programm im Funktionsbaustein		ausgeführtes Programm	
a)						
	:SPA FB 203					
NAME	:BEISP1					
ANNA	:	E 10.3	:U	=ANNA	:U	E 10.3
BERT	:	T 17	:L	KT 010.2	:L	KT 010.2
HANS	:	A 18.4	:SSV	=BERT	:SS	T 17
			:U	=BERT	:U	T 17
			:=	=HANS	:=	A 18.4
b)						
	:SPA FB 204					
NAME	:BEISP2					
MAXI	:	E 10.5	:U	=MAXI	:U	E 10.5
IRMA	:	E 10.6	:SSV	=DORA	:ZV	Z 15
EVA	:	E 10.7	:U	=IRMA	:U	E 10.6
DORA	:	Z 15	:SAR	=DORA	:ZR	Z 15
EMMA	:	M 58.3	:U	=EVA	:U	E 10.7
			:L	KZ 100	:L	KZ 100
			:SVZ	=DORA	:S	Z 15
			:UN	=DORA	:UN	Z 15
			:=	=EMMA	:=	M 58.3
c)						
	:SPA FB 205					
NAME	:BEISP3					
KURT	:	E 10.4	:U	=KURT	:U	E 10.4
KARL	:	T 18	:L	=EGON	:L	EW 20
EGON	:	EW 20	:SVZ	=KARL	:SV	T 18
MAUS	:	M 100.7	:U	=KARL	:U	T 18
			:=	=MAUS	:=	M 100.7

Lade- und Transfer-operationen

Als Aktualoperanden sind die den Grundoperationen entsprechenden Operanden mit Ausnahme der S-Merker zugelassen.

Mit Formaloperanden

Operation	Operand	Funktion
L =		Laden eines Formaloperanden: Der Wert des als Formaloperand vorgegebenen Operanden wird in den AKKU geladen (Parameterart: E, T, Z, A; Parametertyp: BY, W, D).
LC =		Codiertes Laden eines Formaloperanden: Der Wert der als Formaloperand vorgegebenen Zeit- oder Zählzelle wird BCD-codiert in den AKKU geladen (Parametertyp: T, Z).
LW =		Laden des Bitmusters eines Formaloperanden: Das Bitmuster des Formaloperanden wird in den AKKU geladen (Parameterart: D; Parametertyp: KF, KH, KM, KY, KC, KT, KZ).
LD =		Laden des Bitmusters eines Formaloperanden: Das Bitmuster des Formaloperanden wird in den AKKU geladen (Parameterart: D; Parametertyp: KG).
T =	 	Transferieren zu einem Formaloperanden: Der Akkumulatorinhalt wird zu dem als Formaloperand vorgegebenen Operanden transferiert (Parameterart: E,A; Parametertyp: BY, W, D). Formaloperand einsetzen

Bei 'LW=' sind ein Datum in Form eines Binär- (KM) oder eines Hexadezimalmusters (KH), 2 byteweise Betragszahlen (KY), Zeichen (KC), Festpunktzahl (KF), Zeitwert (KT) und Zählwert (KZ) zugelassen.

Bei 'LD=' ist eine Gleitpunktzahl als Datum zugelassen.

mit speziellen Operanden

Operation		Operand		Funktion
	L	BA	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem Bereich "Anschaltung" (BA-Bereich)
		BB	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem erweiterten Bereich "Anschaltung" (BB-Bereich)
	L	BS	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem Bereich "Systemdaten" (BS-Bereich)
		BT	0 bis 255	Laden eines Wortes in den AKKU 1 aus dem erweiterten Bereich "Systemdaten" (BT-Bereich)
	T	BA	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des Bereichs "Anschaltung" (BA-Bereich)
		BB	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des erweiterten Bereichs "Anschaltung" (BB-Bereich)
	T	BS	60 bis 63	Transferieren des Inhalts von AKKU 1 zu einem Wort des Bereichs "Systemdaten" (BS-Bereich)
		BT	0 bis 255	Transferieren des Inhalts von AKKU 1 zu einem Wort des erweiterten Bereichs "Systemdaten" (BT-Bereich)

Im Gegensatz zu den Bereichen BA, BB und BT dürfen vom BS-Bereich nur die Wörter BS 60 bis BS 63 für Anwenderzwecke frei genutzt werden.

Siehe auch das Kapitel "Speicherbelegung und Speicherorgnisation"

Den BT-Bereich können Sie in seiner gesamten Länge (BT 0 bis BT 255) benutzen, vorausgesetzt, Sie verwenden keine Standard-Funktions-bausteine.

Rechen-operationen

Operation		Operand	Funktion
	ENT	-	Es findet ein Stack-Lift in die AKKUs 3 und 4 statt: <AKKU 4> := <AKKU 3> <AKKU 3> := <AKKU 2> <AKKU 2> := <AKKU 2> <AKKU 1> := <AKKU 1> Die AKKUs 1 und 2 werden nicht verändert. Der alte Inhalt des AKKUs 4 geht verloren.

Beispiel Folgender Bruch soll ausgerechnet werden: (30 + 3 * 4) / 6 = 7

	AKKU 1	AKKU 2	AKKU 3	AKKU 4
Vorbelegung der Akkus vor der arithmetischen Operationskette	a	b	c	d
L KF +30	30	a	c	d
L KF +3	3	30	c	d
ENT	3	30	30	c
L KF +4	4	3	30	c
x F	12	30	c	c
+ F	42	c	c	c
L KF +6	6	42	c	c
:F	7	c	c	c

System-operationen Systemoperationen sind in der ersten Spalte der Tabellen mit S gekennzeichnet.

ergänzende arithmetische Operationen

Operation		Operand		Funktion
S	ADD	BF	-128 bis +128	Addieren einer Byte-Konstanten (Festpunkt) zum AKKU-1-L (vorzeichenexpandiert); die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst. - AKKU-1-H sowie die AKKUs 2 bis 4 bleiben unverändert.
S	ADD	KF	-32 768 bis +32 767	Addieren einer Festpunktkonstanten (Wort) zum AKKU-1-L; die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst. - AKKU-1-H sowie die AKKUs 2 bis 4 bleiben unverändert.
S	ADD [1)]	DH	0000 0000 bis FFFF FFFF	Addieren einer Doppelwort-Hexkonstanten zum AKKU 1; die Anzeigen in ANZ 0, ANZ 1, OV und OS werden nicht beeinflusst. - Die AKKUs 2 bis 4 bleiben unverändert.
S	+ D [1)]			Addieren zweier Doppelwort-Festpunktzahlen (AKKU 2 + AKKU 1); das Ergebnis ist über ANZ 0/ANZ 1 auswertbar.
S	- D [1)]			Subtrahieren zweier Doppelwort-Festpunktzahlen (AKKU 2 - AKKU 1); das Ergebnis ist über ANZ 0/ANZ 1 auswertbar.
S	TAK			Tauschen der Inhalte von AKKU 1 und AKKU 2

[1)] Die Programmierung ist abhängig vom PG-Typ und vom Ausgabestand der PG-Systemsoftware.

Organisatorische Operationen

Zu den organisatorischen Operationen gehören auch Systemoperationen.

Hinweis!

Systemoperationen sollten nur von erfahrenen Programmierern und Systemkennern und nur mit Vorsicht angewendet werden.

Systemoperationen sind in der ersten Spalte der Tabellen mit S gekennzeichnet.

Sprung-operationen

Das Sprungziel für unbedingte und bedingte Sprünge wird symbolisch angegeben (maximal 4 Zeichen, beginnend mit einem Buchstaben). Dabei ist der Symbolparameter des Sprungbefehls identisch mit der Symboladresse der anzuspringenden Anweisung.

Bei der Programmierung muss berücksichtigt werden, dass die absolute Sprungdistanz nicht mehr als 127 Wörter umfasst und eine Anweisung aus mehr als einem Wort bestehen kann.

Sprünge dürfen nur innerhalb eines Bausteins durchgeführt werden; Sprünge über "Netzwerke" hinweg sind unzulässig.

Hinweis!

Sprunganweisung und Sprungziel (Symboladresse) müssen im gleichen Netzwerk liegen. Pro Netzwerk darf der Name einer Symboladresse nur einmal vergeben werden.

Ausnahme: Dies gilt nicht für den Sprung SPR, bei dem als Parameter eine absolute Sprungdistanz angegeben wird.

Sprung-operationen

Operation		Operand	Funktion
	SPA = SPB = SPZ =	adr (adr = Symbol-adresse mit maximal 4 Zeichen)	Sprung unbedingt: Der unbedingte Sprung wird unabhängig von Bedingungen ausgeführt. Sprung bedingt: Der bedingte Sprung wird ausgeführt, wenn VKE = 1. Bei VKE = 0 wird die Anweisung nicht ausgeführt und das Verknüpfungsergebnis auf VKE = 1 gesetzt. Sprung bei Ergebnis '0': Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 0 und ANZ 0 = 0 ist. Das Verknüpfungsergebnis wird nicht verändert.
	SPN = SPP = SPM = SPO = SPS =	adr (adr = Symbol-adresse mit maximal 4 Zeichen)	Sprung bei Ergebnis ≠ '0': Der Sprung wird nur dann ausgeführt, wenn ANZ 1 ≠ 0 oder ANZ 0 ≠ 0 ist. Das Verknüpfungsergebnis wird nicht verändert. Sprung bei Ergebnis > '0': Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 1 und ANZ 0 = 0 ist. Das Verknüpfungsergebnis wird nicht verändert. Sprung bei Ergebnis < '0': Der Sprung wird nur dann ausgeführt, wenn ANZ 1 = 0 und ANZ 0 = 1 ist. Das Verknüpfungsergebnis wird nicht verändert. Sprung bei Überlauf (Overflow): Der Sprung wird ausgeführt, wenn die Anzeige OV = 1 ist. Wenn kein Überlauf vorliegt (OV = 0), wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich durch eine arithmetische Operation überschritten wird. Sprung, wenn die Anzeige OS (Overflow speichernd) gesetzt ist: Der Sprung wird ausgeführt, wenn die Anzeige OS = 1 ist. Wenn kein Überlauf vorliegt (OS = 0), wird der Sprung nicht ausgeführt. Das Verknüpfungsergebnis wird nicht verändert. Ein Überlauf entsteht, wenn bei gegebener Zahlendarstellung der zulässige Bereich im Verlauf mehrerer arithmetischer Operationen überschritten wird.
S	SPR	-32768 bis +32767	Beliebiger relativer Sprung innerhalb des Anwenderspeichers bzw. innerhalb eines Funktionsbausteins (z.B. um in ein anderes Netzwerk zu gelangen). Die Operation wird immer ausgeführt, unabhängig von Bedingungen. Als Operand muss die Adressen-Differenz "Sprungziel - aktuelle Operation" (Anzahl Wörter) angegeben werden. Dabei wird der Sprung zu einer höheren (positiver Operand) oder einer niedrigeren (negativer Operand) Adresse als die der aktuellen Operation ausgeführt.

Hinweis!

Bei unsachgemäßer Anwendung der Operation SPR können undefinierte Zustände der Anlage auftreten! Sie sollte daher nur von sehr erfahrenen Programmierern und Systemkennern benutzt werden.

Schiebe-operationen

Bei den Schiebeoperationen ist nur der AKKU 1 an der Ausführung beteiligt.

Der Parameterteil dieser Operation gibt an, um wieviele Stellen der AKKU-Inhalt geschoben bzw. rotiert wird.

Operation	Operand	Funktion
SLW	0 bis 15	Schieben nach links (von rechts werden Nullen nachgezogen)
SRW	0 bis 15	Schieben nach rechts (von links werden Nullen nachgezogen)
SLD	0 bis 31	Schieben eines Doppelwortes nach links (von rechts werden Nullen nachgezogen)
SVW	0 bis 15	Schieben mit Vorzeichen nach rechts (von links wird Bit 15 nachgezogen)
SVD	0 bis 31	Schieben eines Doppelwortes mit Vorzeichen nach rechts (von links wird Bit 31 nachgezogen)
RLD	0 bis 31	Rotieren nach links
RRD	0 bis 31	Rotieren nach rechts

Bei SLW, SRW und SVW ist nur das niederwertige Wort an den Schiebeoperationen beteiligt, bei SLD, SVD, RLD und RRD der gesamte Inhalt des AKKU 1 (32 bit).

Die Schiebeoperationen werden unabhängig von Bedingungen ausgeführt.

Der Wert des zuletzt hinausgeschobenen Bits kann über ANZ 1/Anz 0 mit Sprungoperationen abgefragt werden:

Schieben: letztes geschobenes Bit	ANZ 1	ANZ 0	Sprungoperation
0	0	0	SPZ=
1	1	0	SPN= SPP=

Beispiele

1. Der Inhalt des Datenwortes DW 52 soll gelesen, um 4 Bitstellen nach links verschoben und dann im Datenwort DW 53 abgelegt werden.

Programm:		Inhalt der Datenwörter:
:L	DW 52	KH = 14AF
:SLW	4	
:T	DW 53	KH = 4AF0

2. Das Eingangs-Doppelwort ED 0 soll gelesen und der Inhalt von AKKU 1 so verschoben werden, dass die fettgedruckten Bitstellen des Eingangs-Doppelwortes (z.T. verschoben) erhalten bleiben und die übrigen Bitstellen mit definierten Werten (0 h bzw. 0F h) besetzt werden.

Programm:		Inhalt von AKKU 1 (hexadezimal):	
		AKKU-1-H:	AKKU-1-L:
:L	ED 0	23**48**	A**BC**D
:SLW	4	2348	BCD0
:SRW	4	2348	0BCD
:SLD	4	3480	BCD0
:SVW	4	3480	FBCD
:SVD	4	0348	0FBC
:RLD	4	3480	FBC0
:RRD	4	03**48**	0F**BC**

3. Multiplikation mit 2er-Potenz, z.B. neuer Wert = alter Wert x 8

:L	MW 10	
:SLW	3	
:T	MW 10	**Achtung:** Positive Bereichsgrenze nicht überschreiten!

4. Division durch 2er-Potenz, z.B. neuer Wert = alter Wert : 4

:A	DB 5
:L	DW 0
:SRW	2
:T	DW 0

Umwandlungs-operationen

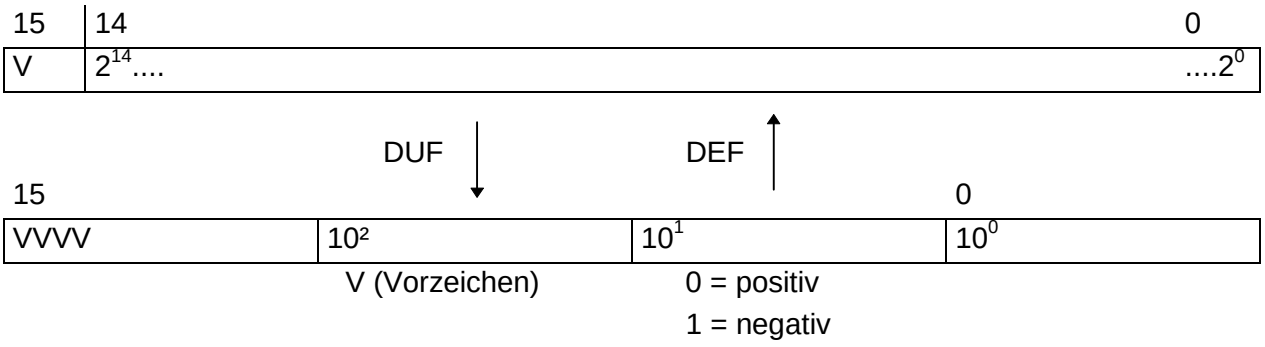
Operation		Funktion
	KEW	Bildung des 1er-Komplements von AKKU-1-L (16 bit)
	KZW	Bildung des 2er-Komplements von AKKU-1-L (16 bit)
	KZD	Bildung des 2er-Komplements von AKKU 1 (32 bit)
	DEF	Festpunktwandlung (16 bit) von BCD in dual
	DUF	Festpunktwandlung (16 bit) von dual in BCD
	DED	Doppelwortwandlung (32 bit) von BCD in dual
	DUD	Doppelwortwandlung (32 bit) von dual in BCD
	FDG	Wandlung einer Festpunktzahl (32 bit) in eine Gleitpunktzahl (32 bit); siehe OB 220: Vorzeichenerweiterung
	GFD	Wandlung einer Gleitpunktzahl in eine Festpunktzahl (32 bit)

DEF

Der im AKKU-1-L (Bit 0 bis Bit 15) stehende Wert wird als BCD-Zahl (Binär-codierte-Dezimalzahl) interpretiert. Nach der Umwandlung steht im AKKU-1-L eine 16-bit-Festpunktzahl.

DUF

Der im AKKU-1-L (Bit 0 bis Bit 15) stehende Wert wird als 16-bit-Fest-punktzahl interpretiert. Nach der Umwandlung steht im AKKU-1-L eine BCD-Zahl.



Hinweis!

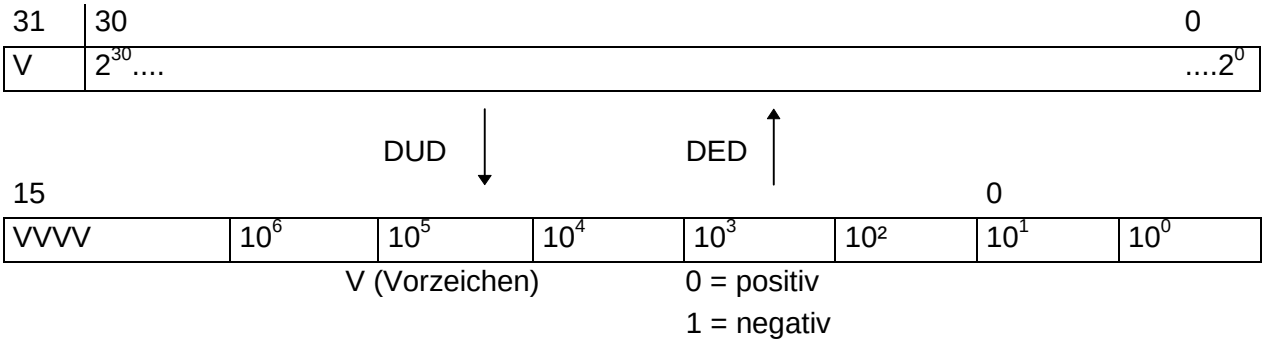
Bitte beachten Sie, dass Ihr Wert in AKKU-1-L vor der Ausführung zwischen -999 und +999 liegt. Ansonsten arbeitet die DUF-Operation fehlerhaft!

DED

Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als BCD-Zahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine 32-bit-Festpunktzahl.

DUD

Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als 32-bit-Festpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine BCD-Zahl.



Hinweis!

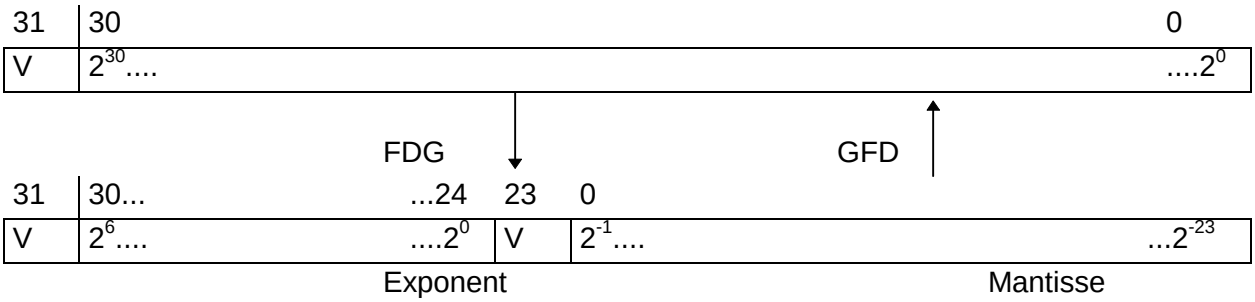
Bitte beachten Sie, dass Ihr Wert in AKKU-1 vor der Ausführung zwischen -9999999 und +9999999 liegt, ansonsten arbeitet die DUD-Operation fehlerhaft!

FDG

Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als 32-bit-Festpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine Gleitpunktzahl (Exponent und Mantisse).

GFD

Der im AKKU 1 (Bit 0 bis Bit 31) stehende Wert wird als Gleitpunktzahl interpretiert. Nach der Umwandlung steht im AKKU 1 eine 32- bit-Festpunktzahl.



Die Umwandlung geschieht durch Multiplikation der (dualen) Mantisse mit dem Wert des (dualen) Exponenten, indem der Mantissenwert über einen gedachten Dezimalpunkt um den Wert des Exponenten (zur Basis '2') zu höherwertigen Bitstellen verschoben wird. Nach der Multiplikation bleiben Reste der ursprünglichen Mantisse rechts vom gedachten Dezimalpunkt übrig. Diese Bitstellen werden vom ganzzahligen Ergebnis abgeschnitten.

Durch diesen Umwandlungsalgorithmus ergeben sich folgende Ergebnisklassen:

- Gleitpunktzahlen ≥ 0 oder ≤ -1 ergeben die nächst kleinere ganze Zahl.
- Gleitpunktzahlen < 0 und > -1 ergeben den Wert '0'.

Umwandlungs-beispiele

Gleitpunktzahl	GFD	32-bit-Festpunktzahl
+5,7	→	5
-2,3	→	-3
-0,6	→	0
+0,9	→	0

Beispiele zu KEW, KZW

1. Der Inhalt des Datenwortes 64 soll Bit für Bit invertiert ("umgekehrt") und in Datenwort 78 abgelegt werden.

Programm:		Inhalt der Datenwörter:
:L	DW 16	KM = 0011111001011011
:KEW		
:T	DW 78	KM = 1100000110100100

2. Der Inhalt des Datenwortes 207 ist als Festpunktzahl zu interpretieren und mit umgekehrten Vorzeichen im Datenwort 51 abzulegen.

Programm:		Inhalt der Datenwörter:
:L	DW 207	KF = +51
:KZW		
:T	DW 51	KF = -51

Dekrementieren/-Inkrementieren

Operation		Operand	Funktion
	D	0 bis 255	Dekrementieren des Low-Bytes (Bits 0 bis 7) von AKKU-1-L um den Operandenwert [1)]
	I	0 bis 255	Inkrementieren des Low-Bytes (Bits 0 bis 7) von AKKU-1-L um den Operandenwert [1)]

[1)] Der Inhalt des Low-Bytes von AKKU-1-L wird um die als Operand angegebene Zahl ohne Übertrag dekrementiert (erniedrigt) bzw. inkrementiert (erhöht). Die Operationsausführung ist unabhängig von Bedingungen.

Beispiel

Programm:		Inhalt der Datenwörter:
:L	DW 7	KH = 1010
:I	16	
:T	DW 8	KH = 1020
:D	33	
:T	DW 9	KH = 10FF

Bearbeitungs-operationen

Operation		Operand		Funktion
	B	DW	0 bis 255	Bearbeite Datenwort: Die nachfolgend angegebene Operation wird mit dem im adressierten Datenwort angegebenen Parameter kombiniert und ausgeführt.
		MW	0 bis 254	Bearbeite Merkerwort: Die nachfolgend angegebene Operation wird mit dem im adressierten M-Merker angegebenen Parameter kombiniert und ausgeführt.
	B=			Bearbeite Formaloperanden (Parameterart: B): Nur A DB, SPA PB, SPA OB, SPA FB, SPA SB können substituiert werden. Formaloperand einsetzen
S	BI		[1]	Indirekte Bearbeitung eines Formaloperanden: Eine Operation ausführen, deren Operationscode in einem Formaloperanden abgelegt ist. Die Nummer des auszuführenden Formaloperanden steht in AKKU 1.
S	B	BS	60 bis 63[1]	Eine Operation, die im Bereich Systemdaten (BS) steht, soll ausgeführt werden (freie Systemdaten: BS 60 bis 63). Bei 2-Wort-Operationen muss das 2. Wort nach BS n+1 geladen werden.

[1] Der Wert, der im Systemdatum oder im Formaloperanden steht, wird als Operationscode einer Operation interpretiert, die dann ausgeführt wird.

Hinweis!

Mit B DW, B MW, BI oder B BS dürfen nur folgende Operationen kombiniert werden:

- ., UN.., O.., ON.., S.., R.., = .. mit den Bereichen E, A, M, S,
- FR T, R T, SA T, SE T, SI T, SS T, SV T;
- FR Z, R Z, S Z, ZR Z, ZV Z,
- L, T mit den Bereichen P, Q, E, A, M, S, D, BA, BB, BS, BT,
- L T, L Z,
- SPA=, SPB=, SPZ=, SPN=, SPP=, SPM=, SPO=,
- SLW, SRW,
- I, SES, SEF,
- A DB, SPA.., SPB.., E DB, EX DX, AX DX, BAB FX, BA FX.

Das PG prüft die Zulässigkeit der Kombinationen nicht!

Beispiele *B DW/B MW*

Operanden-Substitution

Mit den Anweisungen "B DW" und "B MW" können Sie, z.B. in einer Programmschleife, substituiert auf Daten zugreifen. Der substituierte Zugriff setzt sich zusammen aus der Anweisung B DW/B MW und einer unmittelbar nachfolgenden Operation aus dem o.g. Operationsspektrum.

"Substituiert" bedeutet, dass der Operand für die Operation nicht statisch beim Programmieren vorgegeben, sondern erst beim Anlauf Ihres Programms festgelegt wird.

Den Operationstyp wählen Sie beim Programmieren aus dem für die Operation zulässigen Spektrum aus, z.B. PB für die Operation "SPA PB nn".

Den Operandenwert (nn im Beispiel "SPA PB nn") müssen Sie vor einem substituierten Zugriff mit B DW/B MW in ein Daten- oder M-Merkerwort (Parameterwort) laden.

1. Prinzip der Substitution:

```
:L    KF +120
:T    MW 14      MW 14 mit dem Wert "KF +120" laden
:B    MW 14
:L    EB 0
```

vor der Ausführung der Operation "L EB" wird der Operationswert '0' durch den Wert '120' ersetzt; Ausführung: L EB 120

2. Datenwort als Indexregister:

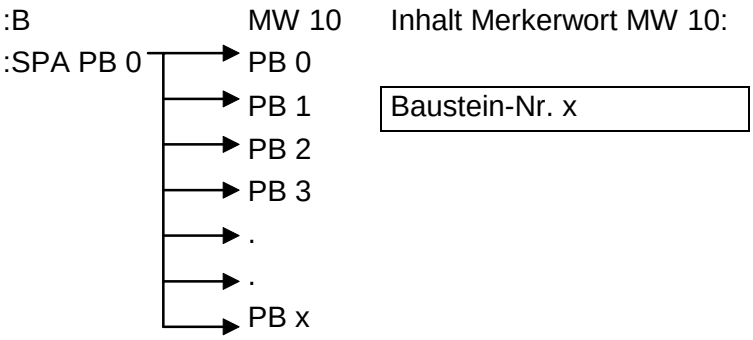
Es sollen die Inhalte der Datenwörter DW 20 bis DW 100 auf den Signalzustand '0' gesetzt werden. Das Indexregister für den Parameter der Datenwörter ist DW 1.

```
      :L    KF +20     Versorgung des Indexregisters
      :T    DW 1
M001  :L    KF +0      Rücksetzen
      :B    DW 1
      :T    DW 0
      :L    DW 1       Erhöhen des Indexregisters
      :L    KF +1
      :+F
      :T    DW 1
      :L    KF +100
      :<=F
      :SPB  =M001      Sprung, wenn Index im Bereich liegt
      ...              weiteres Programm
```

3. Sprungverteiler für Unterprogrammtechnik:

```
                :B      MW 5
                :SPA    M001      Inhalt Merkerwort MW 5:
+               :SPA    M002
Sprung-         :SPA    M003      Sprungdistanz
distanz         :SPA    M004      (maximal ± 127)
                :SPA    M005
                : .
                : .
M001            : .
                : .
                :BEA
M002            : .               Vorteil:
                : .               Alle Teilprogramme stehen
                :BEA              in einem Baustein
M003            : .
                : .
                :BEA
```

4. Sprungverteiler für Bausteinaufrufe:



Operanden-Substitution mit binären Operationen

Bei Operanden-Substitutionen mit binären Operationen können Sie folgende Operandentypen verwenden: Eingänge, Ausgänge, M-Merker, S-Merker, Zeiten und Zähler.

Der Aufbau des M-Merker- oder Datenwortes (Parameterwort) hängt bei dieser Substitution davon ab, welchen Operandentyp Sie verwenden.

Parameterwort für Ein- und Ausgänge

Bit-Nr.	15 11	10 8	7	6 0
	ohne Bedeutung	Bit-Adresse von 0 bis 7	0	Byteadresse von 0 bis 127

Parameterwort für M-Merker

Bit-Nr.	15 11	10 8	7 0
	ohne Bedeutung	Bit-Adresse von 0 bis 7	Byteadresse von 0 bis 255

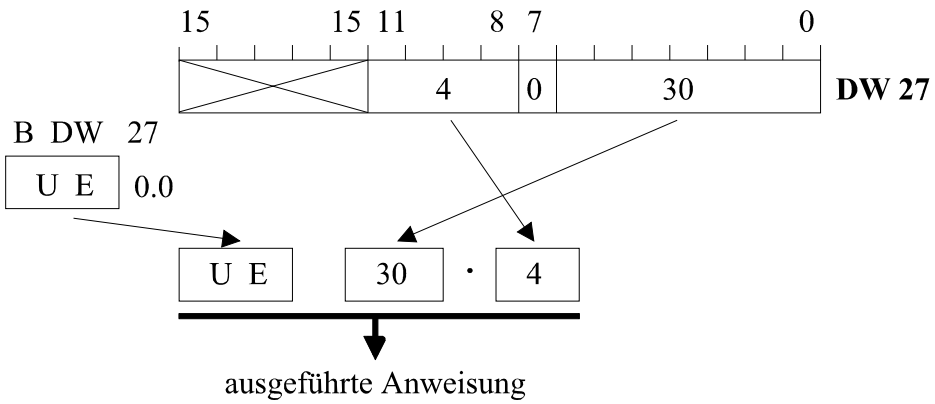
Parameterwort für S-Merker

Bit-Nr.	15	14 12	11 0
		Bit-Adresse von 0 bis 7	Byteadresse von 0 bis 1023

Parameterwort für Zeiten und Zähler

Bit-Nr.	15 8	7 0
	ohne Bedeutung	Nummer von der Zeit- oder Zählerzelle von 0 bis 255

Prinzip der Substitution mit einer binären Operation:



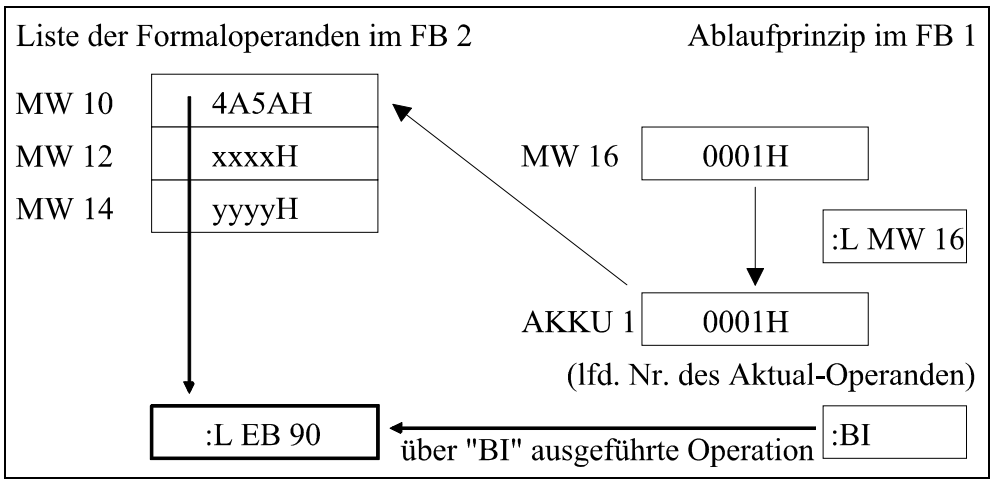
Beispiel zur BI-Operation:

Im Funktionsbaustein FB 1 werden Operationen ausgeführt, deren Operationscodes als Formaloperanden MW 10, MW 12 und MW 14 von einem aufrufenden Baustein übergeben werden.

Welcher der Operationscodes ausgeführt werden soll, wird vom aufrufenden Baustein als lfd. Nummer im Merkerwort MW 16 hinterlegt.

Das Ergebnis der ausgeführten Operation befindet sich anschließend in AKKU 1 und wird in das Merkerwort MW 18 transferiert.

```
FB 1:
NAME    :TEST
BEZ     :MW10     E/A/D/B/Z: D    KM/KH/KY/KC/KF/KT/KZ/KG:KH
BEZ     :MW12     E/A/D/B/Z: D    KM/KH/KY/KC/KF/KT/KZ/KG:KH
BEZ     :MW14     E/A/D/B/Z: D    KM/KH/KY/KC/KF/KT/KZ/KG:KH
        :L        MW 16           lfd. Nr. des Formaloperanden mit dem gewünschten
        :                         Operationscode
        :BI                       übergebener Operationscode wird ausgeführt
        :T        MW 16           Ergebnis in AKKU 1
        :BE
FB 2:
        :L        KF +1
        :T        MW 16           lfd. Nr. des Formaloperanden mit Operationscode
        :SPA      =AUFR
        :
AUFR    :
        :SPA      FB 1            FB TEST aufrufen
NAME    :TEST
MW10    :         KH 4A5A         Operationscode "L EB 90", Formal-Operand 1
MW12    :         KH xxxx         anderer Operationscode, Formal-Operand 2
MW14    :         KH yyyy         anderer Operationscode, Formal-Operand 3
        :T        MW 18
        :BE
```



Prozessalarme sperren/freigeben

"Prozessalarme sperren/freigeben" kann angewendet werden, wenn die prozessalarmgesteuerte Bearbeitung unterdrückt werden soll. In dem Programmteil, der zwischen den Anweisungen AS und AF steht, ist dann die prozessalarmgesteuerte Bearbeitung nicht mehr möglich. Beachten Sie hierzu die Sonderfunktion OB 120 "Alarme gemeinsam sperren".

Operation		Operand	Funktion
	AS		Prozessalarmbearbeitung sperren
	AF		Prozessalarmbearbeitung freigeben

Teil 5 Betriebszustände

Überblick

In diesem Kapitel werden die Betriebszustände STOP, ANLAUF und RUN beschrieben.

Es informiert Sie ausführlich über verschiedene Anlaufarten und damit verbundene Organisationsbausteine, in denen Sie Ihr spezifisches Programm für verschiedene Anlauffälle programmieren können.

Sie erfahren ferner, wodurch sich die Programmbearbeitungen "Zyklische Bearbeitung", "Zeitgesteuerte Bearbeitung" und "Alarmgesteuerte Bearbeitung" auszeichnen und welche Bausteine für Ihr Anwenderprogramm hierbei zur Verfügung stehen.

Nachfolgend sind beschrieben:

- Einführung, Übersicht und Anzeige der Betriebszustände
- Merkmale der Programmbearbeitungsebenen
- Betriebszustand RUN und Urlöschen
- Betriebszustand ANLAUF (Anwenderschnittstelle und Unterbrechungen)
- Betriebszustand RUN - Zyklische Programmbearbeitung
- Betriebszustand RUN - Zeitgesteuerte Programmbearbeitung
- Betriebszustand RUN - Alarmgesteuerte Programmbearbeitung

Inhalt

Thema **Seite**

Teil 5 Betriebszustände 5-1

Einführung und Übersicht 5-2

Programmbearbeitungsebenen 5-5

Betriebszustand STOP 5-8

Betriebszustand ANLAUF 5-11

Betriebszustand RUN 5-15

Einführung und Übersicht

Betriebszustände

Die CPU 928 kennt drei Betriebszustände:

- Betriebszustand STOP
- Betriebszustand ANLAUF
- Betriebszustand RUN

In den Betriebszuständen ANLAUF und RUN können bestimmte Ereignisse auftreten, auf die das Systemprogramm reagieren muss. In vielen Fällen wird dabei ein für das Ereignis vorgesehener Organisationsbaustein (einer von OB 1 bis OB 35) als Anwenderschnittstelle aufgerufen.

Die Betriebszustände werden über LEDs auf der Frontplatte der CPU angezeigt.

Einige der Betriebszustände müssen Sie per Bedienung aktivieren. Dazu können Sie u.a. die Bedienungselemente auf der Frontplatte der CPU benutzen.

LED-Anzeige der Betriebszustände und Fehler

Betriebszustände

Auf der CPU-Frontplatte signalisieren Ihnen verschiedene LEDs den aktuellen Betriebszustand der CPU. Nachfolgende Tabelle zeigt den Zusammenhang zwischen den Anzeigen der STOP- und RUN-LED und dem zugehörigen Betriebszustand.

Diese Anzeigen werden ergänzt durch weitere LED-Anzeigen (BASP, ADF, QVZ, ZYK).

Die Bedeutung der LEDs "RUN" und "STOP" finden Sie in der Tabelle auf der Folgeseite.

Systemfehler

Während des Betriebs erfolgt eine ständige Überwachung von besonders wichtigen und internen Funktionen. Eine Abweichung dieser internen Funktionen vom Normalbetrieb wird erkannt und als Systemfehler behandelt.

Die genauere Spezifikation der Ursache für einen Systemfehler können Sie sowohl dem Bild der LEDs auf der Front entnehmen, als auch dem Systemdatenwort.

Die Bedeutung der LEDs "QVZ", "ADF" und "ZYK" finden Sie in der Tabelle auf der Folgeseite.

Betriebszustände

RUN	STOP	Betriebszustand
an	aus	Die CPU ist im Betriebszustand RUN
aus	an	Die CPU ist im Betriebszustand STOP
aus	aus	Die CPU ist im Betriebszustand ANLAUF oder CPU ist in ANLAUF/RUN, die Bearbeitungskontrolle ist aktiviert und ein Haltepunkt ist erreicht, oder CPU ist in ANLAUF/RUN, die Bearbeitungskontrolle ist aktiviert und ein Haltepunkt wurde vor dem Erreichen wieder ausgetragen (Wartezustand)
aus	blinkt langsam	Die CPU ist im Betriebszustand STOP Die CPU hat einen Übergang in die Betriebsart STOP) verursacht. Ursachen sind z. B.: ADF, QVZ, LZF, BCF, Reglerfehler, Weckfehler, Zyklusfehler, BSTACK-Überlauf, STP-Befehl. Wenn Sie den Betriebsartenschalter auf STOP umlegen, geht das Blinken in Dauerlicht über.
aus	blinkt schnell	Die CPU ist im Betriebszustand STOP Urlöschen wurde angefordert. Diese Anforderung kann von der CPU selbst oder auch per Bedienung erzeugt werden.
an	an	Schwerer Systemfehler Fehlerbehebung: CPU urlöschen; wenn Fehler noch vorhanden, Spannung am AG aus- und wieder einschalten und CPU urlöschen; wenn Fehler noch vorhanden, Spannung am AG ausschalten, CPU ziehen, wieder stecken und urlöschen; wenn Fehler noch vorhanden, kontaktieren Sie die VIPA-Hotline.

Systemfehler

QVZ	ADF	ZYK	Fehlerursache
aus	aus	blinkt schnell	Fehler durch Hardware-Watchdog erkannt
aus	blinkt schnell	aus	Fehler in der internen SP-KP-Kommunikation
aus	blinkt schnell	blinkt schnell	Fehler durch die Überwachung des SPS-Prozessors
blinkt schnell	aus	aus	Fehler durch die Tasküberwachung des Kommunikationsprozessors

LED "BASP"

Sie zeigt an, ob das Bus-Signal BASP (Befehlsausgabe sperren) aktiv ist:

BASP wird von der CPU beim Eintritt in den Zustand RUN inaktiv und beim Übergang in den Zustand STOP aktiv geschaltet.

Im Anlauf und im STOP sowie im ersten Zyklus nach WIEDERANLAUF ist BASP aktiv gesetzt.

Hinweis!

Ist BASP aktiv, so sind alle digitalen Ausgänge gesperrt.

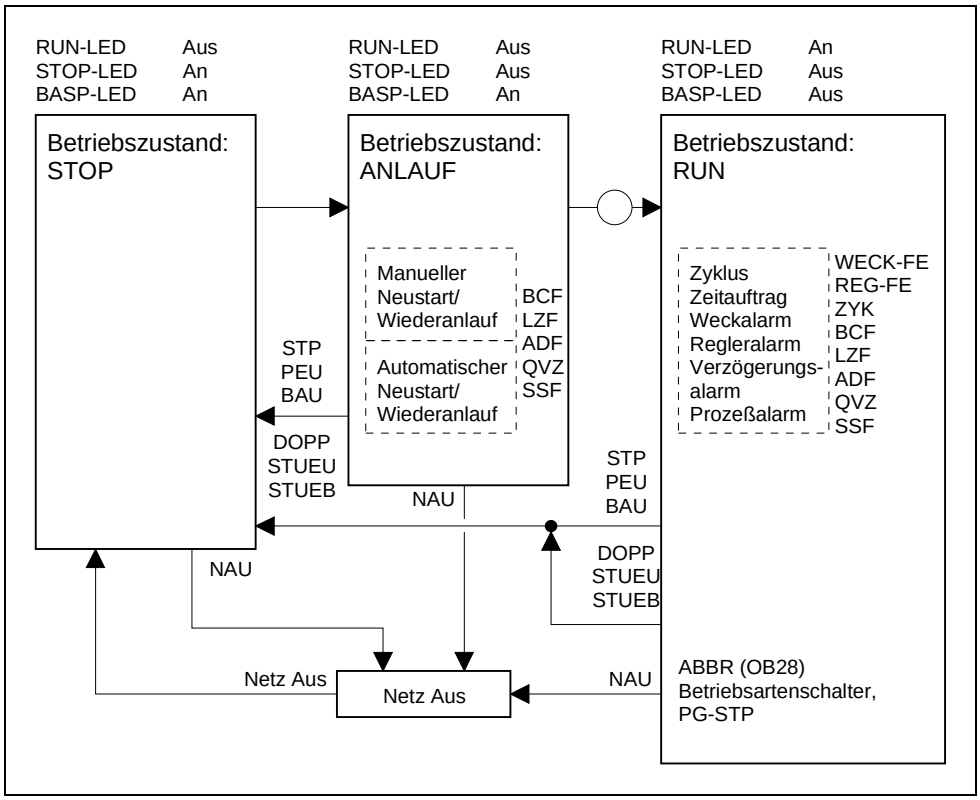
Ist vor dem Übergang in den Betriebszustand RUN ein AUTOMATISCHER oder MANUELLER WIEDERANLAUF durchgeführt worden, erlischt die BASP-LED erst dann, wenn der Restzyklus abgearbeitet worden ist.

LED "PCMCIA"

Die CPU liest Daten von den PCMCIA-Speichermodul. (nur bei Varianten mit PCMCIA-Modulschacht).

Programmbearbeitungsebenen

Nachfolgend finden Sie eine Übersicht der Betriebszustände und Bearbeitungsebenen der CPU 928.



Erläuterung zu den Abkürzungen:

STP =	Stop-Anweisung
PEU =	Ausfall Versorgungsspg. vom Erweiterungsgerät - Peripherie unklar
BAU =	Batterie unklar, Batterie leer oder fehlt
DOPP	=Doppelfehler (Eine noch aktive Fehlerprogrammbearbeitungsebene wird ein zweites Mal aktiviert - Neustart erforderlich)
STUEU	=USTACK - Überlauf
STUEB	=BSTACK - Überlauf
NAU =	Netzausfall
PG-STP	=Stop Anwahl am PG

Programm-bearbeitungs-ebenen im ANLAUF

- Manueller Neustart
- Manueller Wiederanlauf
- Automatischer Neustart
- Automatischer Wiederanlauf

→ Anlaufebene

- BCF (Befehlscodefehler)
- LZF (Laufzeitfehler)
- ADF (Adressierungsfehler)
- QVZ (Quittungsverzug)
- SSF (Schnittstellenfehler)

→ Fehlerebene

Programm-bearbeitungs-ebenen im RUN

- Zyklus
- Zeitauftrag
- Weckalarm 5s (zeitgesteuert)
- Weckalarm 2s (zeitgesteuert)
- Weckalarm 1s (zeitgesteuert)
- Weckalarm 500ms (zeitgesteuert)
- Weckalarm 200ms (zeitgesteuert)
- Weckalarm 100ms (zeitgesteuert)
- Weckalarm 50ms (zeitgesteuert)
- Weckalarm 20ms (zeitgesteuert)
- Weckalarm 10ms (zeitgesteuert)
- Regleralarm (zeitgesteuert)
- Verzögerungsalarm (zeitgesteuert)
- Prozessalarm (alarmgesteuert)

→ Grundebene

- WECK-FE (Weckfehler)
- REG-FE (Reglerfehler)
- ZYK (Zyklusfehler)
- BCF (Befehlscodefehler)
- LZF (Laufzeitfehler)
- ADF (Adressierungsfehler)
- QVZ (Quittungsverzugfehler)
- SSF (Schnittstellenfehler)
- ABBR (Abbruch)

→ Fehlerebene

Merkmale einer Programmbearbeitungsebene

Eine Programmbearbeitungsebene ist durch bestimmte Merkmale charakterisiert, die hier kurz zusammengefasst sind.

Einschachtelung von anderen Ebenen

Bei einem Ereignis, das eine höherpriore Bearbeitung erfordert, wird die aktuelle Ebene vom Systemprogramm unterbrochen und die höherpriore Ebene eingeschachtelt.

Spezifisches Systemprogramm

Jede Programmbearbeitungsebene hat ihr eigenes Systemprogramm.

USTACK

Nach dem Aufruf eines OBs werden die Anweisungen ausgeführt. Hierbei wird der aktuelle Registersatz in den USTACK gerettet und ein neuer Registersatz angelegt.

Ein Registersatz besteht aus: AKKU 1...4, Bausteinstack-Pointer, Baustein-Adressregister, Datenbaustein-Anfangsadresse/-Länge, Adresszähler und Basisadressregister.

Priorität

Die Programmbearbeitungsebenen besitzen eine feste Priorität. Abhängig von der Priorität können sich diese gegenseitig unterbrechen bzw. ineinander verschachtelt werden.

Grundebenen können sich nur an Bausteingrenzen einschachteln, Anlauf und Fehlerebenen werden, sobald das Ereignis aufgetreten ist, an der nächsten Befehlsgrenze eingeschachtelt.

Bei Fehlern hat der zuletzt aufgetretene immer höchste Priorität.

Reaktion bei einem Doppelfehler

Eine einmal aktivierte Fehlerebene, die noch nicht vollständig abgearbeitet ist, kann nicht nochmals aktiviert werden. In diesem Fall geht die CPU in unmittelbaren STOP. Eine Ausnahme bilden hier die Weckfehler.

Im USTACK mit Tiefe "01" ist die Kennung DOPP und die aufgerufene Fehlerebene angekreuzt.

Betriebszustand STOP

Kennzeichen

Der Betriebszustand STOP ist durch folgende Merkmale charakterisiert:

Anwenderprogramm

Das Anwenderprogramm wird nicht bearbeitet.

Erhalt von Daten

Hat zuvor eine Programmbearbeitung stattgefunden, dann bleiben die Werte von Zählern, Zeiten, Merkern und des Prozessabbilds beim Übergang in den STOP-Zustand erhalten.

BASP-Signal

Die Befehlsausgabe ist gesperrt d.h. alle digitalen Ausgaben sind gesperrt.

USTACK

Hat zuvor eine Programmbearbeitung stattgefunden, so finden Sie im USTACK einen Informationsblock mit der Unterbrechungsursache.

LED-Anzeigen

RUN-LED	Aus
BASP-LED	An (außer im Testbetrieb)

STOP-LED An

Betriebszustand STOP wurde ausgelöst durch:

- RUN/STOP-Schalter auf STOP
- PG-Funktion AG-STOP
- PG-Funktion "Bearbeitungskontrolle Ende"
- Gerätefehler (BAU, PEU)
- Urlöschen

STOP-LED langsames Blinken

Betriebszustand STOP wurde ausgelöst durch:

- Stoppbefehl in Anwenderprogramm
- Fehlerbedienung (DB1-Fehler, Anlaufartfehler)
- Programmier bzw. Gerätefehler (zusätzlich leuchten ADF, QVZ, ZYK)
- PG-Funktion "Bearbeitungskontrolle Ende"

STOP-LED schnelles Blinken (2Hz)

Ein schnelles Blinken signalisiert, dass URLÖSCHEN angefordert ist.

Urlöschen

Das Urlöschen bringt die CPU wieder in den Auslieferungszustand. Programm, Daten, Register und Zeiger werden gelöscht und falls ein EPROM bzw. eine Flash-Card gesteckt ist, werden nach dem Löschen die Daten in die CPU übertragen. Bevor das Urlöschen ausgeführt werden kann, muss es angefordert werden. Bei Betrieb mit EPROM ist ein Neustart mit Rücksetzen möglich.

Urlöschen anfordern

Urlöschen kann vom System oder vom Anwender angefordert werden.

Anforderung vom Systemprogramm

Nach jedem Einschalten der Netzspannung und nach dem Urlöschen wird eine Initialisierungsroutine durchlaufen. Wird hierbei ein Fehler bemerkt, so geht die CPU in den STOP-Zustand mit schnellem Blinken über.

Die Störursache ist dann in jedem Fall zu beseitigen. Sollte dies nicht möglich sein, dann wenden Sie sich an die VIPA-Hotline.

Anforderung vom Anwender

Durch folgende Bedienschritte fordern Sie das Urlöschen an:

Bringen Sie den RUN/STOP-Schalter in Stellung STOP (STOP-LED zeigt Dauerlicht)

Halten Sie den RESET-Taster in Stellung "Overall" gedrückt und schalten Sie den RUN/STOP-Schalter von STOP auf RUN und wieder zurück auf STOP. Daraufhin zeigt die STOP-LED schnelles Blinken. Das Urlöschen ist jetzt angefordert. Sie können nun den RESET-Taster loslassen.

Hinweis!

Wenn Sie nun doch nicht löschen möchten, dann führen Sie einen Neustart oder Wiederanlauf durch.

Urlöschen durchführen

Das Urlöschen kann nur von Ihnen durchgeführt werden.

Zum Urlöschen der CPU halten Sie den RESET-Taster in Stellung "Overall" gedrückt und schalten Sie den RUN/STOP-Schalter von STOP auf RUN und wieder zurück auf STOP. Daraufhin zeigt die STOP-LED Dauerlicht. Das Urlöschen ist jetzt durchgeführt. Sie können nun den RESET-Taster loslassen.

Sie können die CPU auch vom PG aus löschen. Verwenden Sie hierzu die PG-Funktion URLÖSCHEN. Die manuelle Löschanforderung kann entfallen. Die Stellung des Betriebsartenschalters ist hierbei irrelevant.

Hinweis!

Nach dem Urlöschen ist nur die Anlaufart NEUSTART zulässig.

Prüfsumme

Beim Urlöschen wird eine Prüfsumme über das Systemprogramm gebildet und mit dem Eintrag im EPROM verglichen. Stimmen beide nicht überein, liegt ein schwerer Systemfehler vor.

Laden der Memory Card

Ist beim Durchführen des Urlöschens eine Memory Card gesteckt, so werden alle dort vorhandenen Code- und Datenbausteine in den Anwenderspeicher der CPU geladen. Danach befindet sich die CPU im "EPROM-Betrieb", d.h. Codebausteine können nicht nachgeladen, verändert oder gelöscht werden, ein Nachladen, Verändern und Löschen von Datenbausteinen im DB-RAM ist dagegen möglich.

Betriebszustand ANLAUF

Kennzeichen

Folgende Eigenschaften sind kennzeichnend für die Betriebsart Anlauf:

Organisationsbausteine (OBs)

Die Länge der Organisationsbausteine ist nicht beschränkt. Auch wird der Ablauf zeitlich nicht überwacht. In den Anlauf OBs können weitere Bausteine aufgerufen werden.

Folgende Organisationsbausteine werden aufgerufen:

Manueller oder automatischer Neustart:	OB 20,
Manueller Wiederanlauf	OB 21,
Automatischer Wiederanlauf	OB 22.

Datenbehandlung

Die Werte von Zählern, Zeiten, Merkern und Prozessabbildern werden in den einzelnen Anlaufarten unterschiedlich behandelt.

BASP-Signal

Beim Anlauf sind alle digitalen Ausgaben gesperrt d.h. die Befehlsausgabesperre ist aktiv.

LEDs auf der Frontplatte

RUN-LED	aus
STOP-LED	aus
BASP-LED	an

Anlaufarten

In der Betriebsart Anlauf gibt es folgende zwei grundsätzlichen Arten des Anlaufs, die wiederum untergliedert sind:

- Neustart,
- Wiederanlauf.

Neustart

Alle Merker, Zeiten, Zähler und das Prozessabbild werden gelöscht. Die Bearbeitung des Anwenderprogramms beginnt von vorn.

Ein Neustart ist immer zulässig, sofern vom Systemprogramm nicht Urlöschen angefordert wird.

Neustart mit Rücksetzen von Hand (manueller Neustart)

Für den manuellen Neustart halten Sie den RESET-Taster in Stellung RESET und bringen Sie den RUN/STOP-Schalter in Stellung RUN. Die CPU führt daraufhin einen manuellen Neustart durch und zeigt mit der RUN-LED den RUN-Zustand an. Sie können nun den RESET-Taster loslassen.

Automatischer Neustart

Nach Wiederkehr der Versorgungsspannung geht die CPU selbständig wieder in den Betriebszustand. Alle Merker, Zeiten, Zähler und das Prozessabbild werden gelöscht. Die Bearbeitung des Anwenderprogramms beginnt von vorn.

Nur wenn dies im DX 0 parametriert wurde.

Wiederanlauf

Die Zustände der Merker, Zeiten, Zähler und das Prozessabbild bleiben während der Stillstandszeit erhalten. Die Bearbeitung des Anwenderprogramms wird an der Unterbrechungsstelle fortgesetzt.

Wiederanlauf ist nicht zulässig, wenn Urlöschen angefordert ist oder eins der folgenden Ereignisse eingetreten ist:

- Doppelaufruf einer Programmbearbeitungsebene,
- Urlöschen (Steuerbits: URGELOE),
- Anlaufabbruch (Steuerbits: ANL_ABB),
- STOP nach PG-Funktion BEARBEITUNGSKONTROLLE ENDE,
- Komprimieren im STOP,
- Stacküberlauf,
- Änderung des Anwenderprogramms im STOP.

Wiederanlauf von Hand (manuell)

Für den manuellen Wiederanlauf bringen Sie (ohne den RESET-Taster zu betätigen) den RUN/STOP-Schalter in Stellung RUN. Sie können auch die PG-Funktion AG-START (WIEDERANLAUF) verwenden.

Wiederanlauf automatisch (Standard)

Nach Wiederkehr der Versorgungsspannung geht das Gerät selbständig wieder in den Betriebszustand RUN über, sofern der DX0 entsprechend parametriert ist.

Voraussetzung für den automatischen Wiederanlauf:

- RUN/STOP-Schalter befindet sich noch immer in Stellung RUN
- CPU befindet sich vor dem Spannungsausfall im RUN
- Bei der Initialisierung sind keine Fehler aufgetreten

Anwender-schnittstellen für den Anlauf

Als Anwenderschnittstellen für die verschiedenen Anlaufarten dienen die Organisationsbausteine OB 20, OB 21 und OB 22. In diesen Bausteinen können Sie Ihr Programm für die verschiedenen Anlaufarten hinterlegen. In den Anlauf-OBs können Sie z.B. Merker setzen, Zeiten starten, Datenverkehr CPU mit Peripherie vorbereiten und die Synchronisation von mehreren CPs durchführen.

Die OBs haben folgende Funktionen:

OB 20 - Neustart

Sobald die CPU einen automatischen oder manuellen Neustart durchführt, wird vom Systemprogramm einmal der OB 20 aufgerufen. Legen Sie in OB 20 ein Programm ab, das vorbereitende Schritte für einen Neustart der zyklischen Bearbeitung durchführt.

Nach der Bearbeitung des OB 20 beginnt die zyklische Programmbearbeitung durch Aufruf von OB 1 bzw. FB 0.

OB 21 - Manueller Wiederanlauf

Sobald die CPU einen manuellen Wiederanlauf durchführt, ruft das Systemprogramm OB 21 auf. Legen Sie in OB 21 ein Programm ab, das den Wiederanlauf eines zuvor im RUN befindlichen Programms regelt. Nach dem OB 21 wird die zyklische Programmbearbeitung an der Abbruchstelle aufgerufen und mit der nächsten Anweisung fortgesetzt.

Hierbei gilt:

- Das BASP-Signal bleibt während der Bearbeitung des Restzyklus aktiv und wird erst deaktiviert, wenn ein neuer Zyklus beginnt.
- Das Prozessabbild bleibt am Ende des Restzyklus zurückgesetzt.

Hinweis!

Auch im STOP registriert die CPU einen Netzspannungsausfall. Wenn Sie danach einen manuellen Wiederanlauf auslösen, so ruft die CPU vor dem OB 21 den OB 22 auf. Bei manuellem Neustart ignoriert die CPU das vorangegangene und ruft direkt den OB 20 auf.

OB 22 - Automatischer Wiederanlauf

Bei automatischem Wiederanlauf ruft das Systemprogramm den OB 22 auf. Legen Sie in OB 22 ein Programm ab, das den Wiederanlauf eines zuvor im RUN befindlichen Programms regelt. Nach dem OB 22 wird die zyklische Programmbearbeitung an der Abbruchstelle aufgerufen und mit der nächsten Anweisung fortgesetzt.

Hierbei gilt wie beim manuellen Wiederanlauf:

- Das BASP-Signal bleibt während der Bearbeitung des Restzyklus aktiv und wird erst deaktiviert, wenn ein neuer Zyklus beginnt.
- Das Prozessabbild bleibt am Ende des Restzyklus zurückgesetzt.

Unterbrechungen im Anlauf

Ein Anlaufprogramm kann unterbrochen werden durch

- Netzspannungsausfall im Zentralgerät oder im Erweiterungsgerät
- STOP (Schalter, Befehl)
- Programm- und Gerätefehler

Netzspannungsausfall im Anlauf

Bei Netzspannungsausfall im Anlauf sind abhängig von der gewählten Betriebsart mehrere Fälle zu unterscheiden:

Netzspannungsausfall bei ...	Betriebsart Automatischer Wiederanlauf	Betriebsart Automatischer Neustart
Neustart (OB 20)	OB 22 wird an Unterbrechungsstelle im OB 20 eingeschachtelt	OB 20 wird abgebrochen und neu aufgerufen
Manuellem Wiederanlauf (OB 21)	OB 22 wird an Unterbrechungsstelle im OB 21 eingeschachtelt	OB 21 wird abgebrochen und neu aufgerufen
Automatischem Wiederanlauf (OB 22)	OB 22 wird abgebrochen und neu aufgerufen	OB 22 wird abgebrochen und neu aufgerufen

Betriebszustand RUN

Merkmale

Wenn die CPU einen Anlauf fertig bearbeitet hat, geht Sie in den Betriebszustand RUN über. Folgende Merkmale charakterisieren diesen Zustand:

Bearbeitung des Anwenderprogramms

Das Anwenderprogramm im OB 1 bzw. im FB 0 wird zyklisch bearbeitet, wobei zusätzlich alarmgesteuert weitere Programmteile eingeschachtelt werden können.

Zeiten, Zähler, Prozessabbild

Alle im Programm gestarteten Zeiten und Zähler laufen und das Prozessabbild wird zyklisch aktualisiert.

BASP-Signal

Das BASP-Signal (Befehlsausgabesperre) wird deaktiviert d.h. alle digitalen Ausgänge sind freigegeben.

LEDs auf der Frontplatte

RUN-LED	an
STOP-LED	aus
BASP-LED	aus

Hinweis!

Ist vor dem Übergang in den Betriebszustand RUN ein automatischer oder manueller Wiederanlauf durchgeführt worden, erlischt die BASP-LED erst dann, wenn der Restzyklus abgearbeitet und das Prozessabbild aktualisiert worden ist.

Sie können den Betriebszustand RUN nur über ANLAUF erreichen.

Programmbearbeitungsebenen

Die Programmbearbeitungsebenen werden durch unterschiedliche Ereignisse ausgelöst. Für jede Bearbeitungsebene stehen Ihnen als Anwenderschnittstelle eigene OBs und FBs zur Verfügung. In der CPU 928 können alle Grundprogrammbearbeitungsebenen gleichzeitig programmiert sein.

Abhängig von der Priorität erfolgt der Aufruf der entsprechenden Ebene aus dem Systemprogramm heraus.

Im Betriebszustand RUN bietet Ihnen die CPU folgende Grundprogrammbearbeitungsebenen:

- Zyklus
  Das Anwenderprogramm wird zyklisch bearbeitet.
- Zeitauftrag
  In einem definierbaren Zeitintervall wird Ihr Anwenderprogramm bearbeitet. Sie können aber auch einen festen Zeitpunkt vorgeben, zu dem das Anwenderprogramm einmalig bearbeitet werden soll.
- 9 Weckalarme

  Das Anwenderprogramm wird in festen, vom System angebotenen Zeitintervallen bearbeitet.
- Prozessalarm
  Das Anwenderprogramm wird alarmgesteuert bearbeitet.

Zyklische Programm-bearbeitung

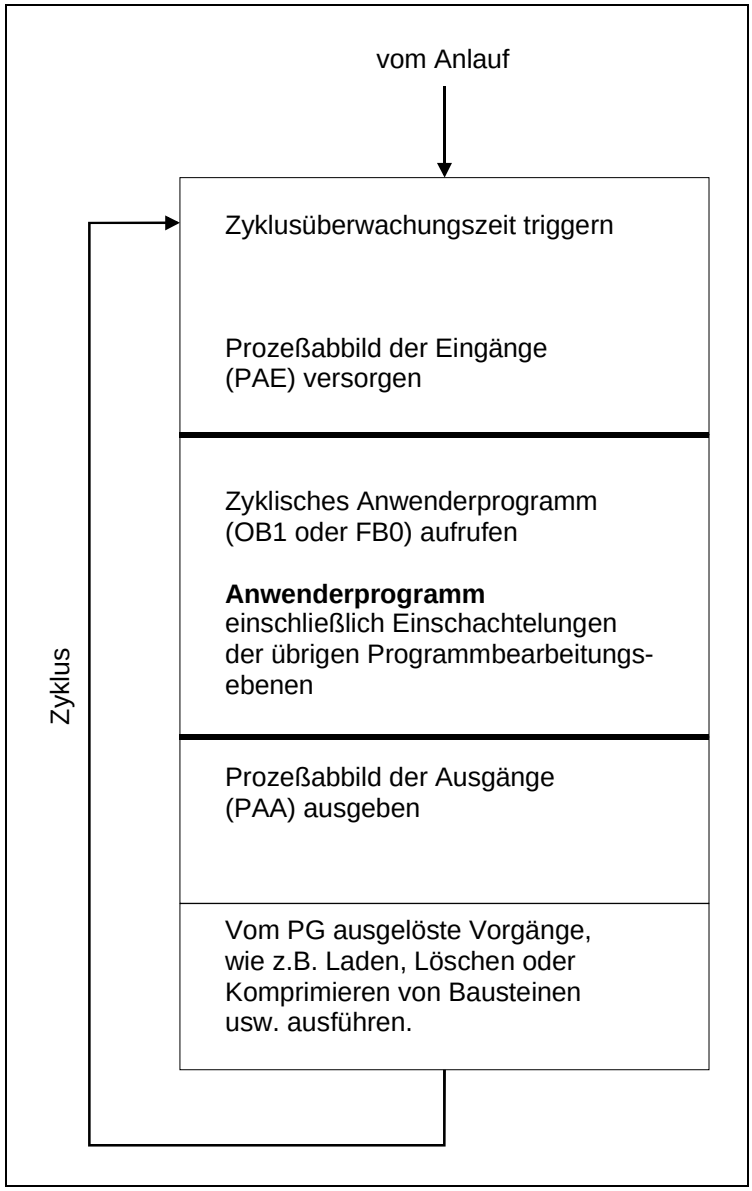
Im Allgemeinen wird bei der SPS die zyklische Programmbearbeitung eingesetzt (Programmbearbeitungsebene ZYKLUS). Es handelt sich hierbei um einen freien Zyklus. Nach Erreichen des Zyklusendes beginnt sofort die nächste zyklische Bearbeitung.

Auslösung

Hat die CPU das Anlaufprogramm fehlerfrei abgearbeitet, so beginnt sie mit der zyklischen Programmbearbeitung.

Prinzip

Das Prinzip der zyklischen Programmbearbeitung soll folgende Abbildung verdeutlichen:


vom Anlauf
Zyklusüberwachungszeit triggern
Prozeßabbild der Eingänge (PAE) versorgen
Zyklisches Anwenderprogramm (OB1 oder FB0) aufrufen
Anwenderprogramm einschließlich Einschachtelungen der übrigen Programmbearbeitungs-ebenen
Prozeßabbild der Ausgänge (PAA) ausgeben
Vom PG ausgelöste Vorgänge, wie z.B. Laden, Löschen oder Komprimieren von Bausteinen usw. ausführen.
Zyklus

Anwenderschnittstelle OB1 bzw. FB0

Bei der zyklischen Programmbearbeitung wird als Anwenderschnittstelle regelmäßig der OB 1 oder der FB 0 aufgerufen.

Das Anwenderprogramm im OB 1 bzw. FB 0 wird von Anfang an über die darin von Ihnen programmierten Bausteinaufrufe hinweg durchgehend bearbeitet.

Unterbrechungsstellen

Die zyklische Programmbearbeitung kann an den Bausteingrenzen unterbrochen werden durch:

- prozessalarmgesteuerte Programmbearbeitung,
- Reglerbearbeitung,
- zeitgesteuerte Programmbearbeitung.

Hinweis!

Durch Parametrierung des DX0 können diese Unterbrechungen auch an Befehlsgrenzen erfolgen.

Die zyklische Programmbearbeitung kann an den Befehlsgrenzen unterbrochen bzw. ganz abgebrochen werden durch:

- Auftreten eines Geräte- oder Programmfehlers,
- Bedienung (PG-Funktion, STOP-Schalter),
- STOP-Befehl.

AKKUs als Datenspeicher

Sie können bei der CPU 928 die Rechenregister AKKU 1 bis 4 über Zyklusgrenzen hinweg (Ende Programmzyklus bis zum Beginn des nächsten) als Datenspeicher verwenden.

Zeitgesteuerte Programmbearbeitung

Bei einer zeitgesteuerten Programmbearbeitung wird die CPU von einem internen Takt oder von einem Zeitsignal veranlasst, die aktuelle Programmbearbeitung zu unterbrechen und ein spezifisches Programm zu bearbeiten. Nach der Bearbeitung kehrt die CPU wieder zurück an die Unterbrechungsstelle. Auf diese Weise können Sie zeitgesteuert Programmabschnitte einschieben.

Bei der zeitgesteuerten Programmbearbeitung stehen Ihnen unterschiedliche Arten der Auslösung zur Verfügung:

Urzeitgesteuerter Weckalarm

- Eine Auslösung erfolgt nach einem frei gewählten Zeitraster oder einmalig zu einem absoluten Zeitpunkt. Sobald der eingestellte Zeitpunkt erreicht wird, wird OB 9 aufgerufen.
- Weckalarme

  Auslösung in 9 verschiedenen Zeitrastern 10ms ... 5s durch Weckalarme. Jedem Weckalarm ist ein OB zugeordnet (OB 10 ... OB 18). Hierbei handelt es sich um feste Zyklen, d.h. die Zeitdauer zwischen zwei Programmstarts ist fest.

Achtung!

Bei der zeit- oder alarmgesteuerten Bearbeitung besteht die Gefahr, dass Merker überschrieben werden. Sie können die Merker zu Beginn einer alarm- bzw. zeitgesteuerten Bearbeitung retten und später wieder zurückschreiben oder S-Merker verwenden, die in genügender Anzahl verfügbar sind.

Urzeitgesteuerter Weckalarm

Die auf der CPU 928 befindliche Hardwareuhr können Sie per Programm setzen und auslesen. Über diese Uhr können Sie Programmteile zeitgesteuert bearbeiten. Dieser Alarm eignet sich besonders zur Bearbeitung von einmaligen Vorgängen oder Vorgängen, die in großen Zeitabständen zyklisch auftreten.

Auslösung

Sie erzeugen einen uhrzeitgesteuerten Weckalarm durch Aufruf des Sonder-OBs OB 151. Ist die eingestellte Zeit oder das Datum erreicht, so wird der Zeitauftrag bearbeitet. Dies kann einmalig geschehen (Absolutzeit) oder sich wiederholen (Zeitraster). Sobald ein eingestellter Zeitpunkt erreicht wird, wird der OB 9 aufgerufen. Nach der Bearbeitung des OB 9 erfolgt wieder ein Rücksprung an die Unterbrechungsstelle und das Anwenderprogramm wird wieder fortgesetzt.

Beispiel: Sie möchten z.B. jede Minute beim Sekundenwert 40 einen Weckalarm auslösen.

Einstellungen über den OB 151:

```
SEKUNDEN:          40
AUFTRAGSART:       1 (minütlich)
```

Anwenderschnittstelle OB 9

Als Anwenderschnittstelle wird bei einem uhrzeitgesteuerten Weckalarm der OB 9 aufgerufen. Im OB 9 hinterlegen Sie Ihr Programm, das dann abgearbeitet werden soll. Ist kein OB 9 vorhanden, dann wird das Anwenderprogramm nicht unterbrochen.

Besonderheiten

Ein uhrzeitgesteuerter Weckalarm wird nur im RUN bearbeitet. Uhrzeitgesteuerte Weckalarme, die nicht im RUN auftreten, werden verworfen.

Ein nach Urlöschen und Neustart erzeugter uhrzeitgesteuerter Weckalarm bleibt bei einem WIEDERANLAUF und über NETZ AUS / NETZ EIN erhalten.

Es ist immer nur ein uhrzeitgesteuerter Weckalarm gültig.

Wenn Sie einen neuen uhrzeitgesteuerten Weckalarm erzeugen, wird ein zuvor definierter uhrzeitgesteuerter Weckalarm storniert.

Tritt ein uhrzeitgesteuerter Weckalarm auf, ohne dass der vorangegangene vollständig bearbeitet ist, so wird der neue Weckalarm verworfen.

Weckalarme

In der CPU 928 können Sie bis zu 9 verschiedene Programme zeitgesteuert bearbeiten lassen, von denen jedes in einem anderen Zeitraster aufgerufen wird.

Auslösung

Ein Weckalarm wird automatisch in seinem Zeitraster ausgelöst, wenn der entsprechende OB geladen ist.

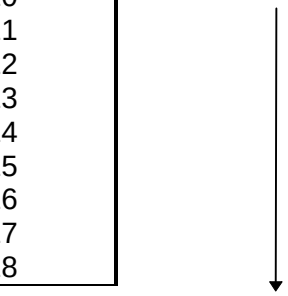
Anwenderschnittstellen OB 10 ... OB 18

Als Anwenderschnittstelle wird bei Auftreten eines bestimmten Weckalarms an der nächsten Bausteingrenze (bzw. Befehlsgrenze) der zugehörige OB eingeschachtelt.

Folgende Tabelle zeigt die Zuordnung der Weckalarmzeiten zu den OBs:

Zeitraster	aufgerufener OB
10ms	OB 10
20ms	OB 11
50ms	OB 12
100ms	OB 13
200ms	OB 14
500ms	OB 15
1s	OB 16
2s	OB 17
5s	OB 18

Fallende Priorität



Sie möchten z.B., dass alle 500ms ein Programm in die zyklische Programmbearbeitung eingeschoben werden soll. Legen Sie hierzu Ihr einzuschiebendes Programm im OB 15 ab.

Hinweis!

OBs mit kürzeren Zeitrastern sind höherprior und können OBs mit längeren Zeitrastern unterbrechen!

Zeit seit letzter Alarmbearbeitung

Bei jedem Aufruf eines Weckalarm-OBs hinterlegt die CPU im AKKU 1, wieviele Zeitraster seit dem letzten Aufruf des Weckalarm-OBs aufgetreten sind.

Hierbei gilt: AKKU 1 := Anzahl Zeitraster

Steht bei Aufruf des OB 11 beispielsweise die Zahl "3" in AKKU 1, bedeutet dies, dass seit dem letzten Aufruf des OB 11 3 Zeitraster (3 x 20ms) vergangen sind. Solange kein Weckfehler vorliegt, wird an AKKU 1 eine "0" übergeben.

Weckfehler (WECK-FE)

Sobald während der Bearbeitung eines Weckalarm-OBs der gleiche Alarm nochmals bearbeitet werden soll, liegt ein Weckfehler vor. Ein Weckfehler liegt auch vor, wenn ein Weckalarm-OB noch nicht bearbeitet werden konnte und dieser Alarm erneut auftritt. Dies ist nur mit der Einstellung "Weckalarme unterbrechen an Bausteingrenzen" möglich. Je schneller eine zeitgesteuerte Programmbearbeitungsebene ist, um so größer ist die Gefahr von Weckfehlern.

Bei einem Weckfehler ruft das Systemprogramm den OB 33 (WECK-FE) auf. Im OB 33 können Sie mit einem Programm auf den Weckfehler reagieren. Ist der OB 33 nicht geladen, geht die CPU in den STOP-Zustand und im USTACK ist WECK-FE angekreuzt und die entsprechende Ebene angegeben.

Beim Aufruf des OB 33 hinterlegt das Systemprogramm in AKKU 1 und AKKU 2 zusätzlich Informationen, die den ersten aufgetretenen Fehler näher erläutern:

Fehlerkennung		Erläuterung
AKKU-1-L	AKKU-2-L	
1001h	0016h	Weckfehler bei OB 10 (10ms)
1001h	0014h	Weckfehler bei OB 11 (20ms)
1001h	0012h	Weckfehler bei OB 12 (50ms)
1001h	0010h	Weckfehler bei OB 13 (100ms)
1001h	000Eh	Weckfehler bei OB 14 (200ms)
1001h	000Ch	Weckfehler bei OB 15 (500ms)
1001h	000Ah	Weckfehler bei OB 16 (1s)
1001h	0008h	Weckfehler bei OB 17 (2s)
1001h	0006h	Weckfehler bei OB 18 (5s)

Die Kennung im AKKU-2-L ist die Ebenenkennung des fehlererzeugenden Weckalarms.

Programmbearbeitung fortsetzen

Soll die Programmbearbeitung bei einem aufgetretenen Weckfehler fortgesetzt werden, programmieren Sie einfach im OB 33 die Anweisung "BE". Nach der Bearbeitung des OB 33 wird die Programmbearbeitung fortgesetzt.

Sie können aber auch im DX0 die Voreinstellung ändern, dass bei einem aufgetretenen Weckfehler die Programmbearbeitung fortgesetzt wird.

Hinweis!

Bitte beachten Sie die Sonderfunktionen OB 120 ... OB 123 mit denen Sie die Bearbeitung von Weckalarmen für einen bestimmten Programmteil sperren bzw. verzögern können.

Alarmgesteuerte Programmbearbeitung (Prozessalarm)

Eine alarmgesteuerte Programmbearbeitung liegt vor, wenn ein Bus-Signal einer interruptfähigen Baugruppe die CPU veranlasst, die Programmbearbeitung zu unterbrechen und ein spezifisches Programm zu bearbeiten. Nach der Bearbeitung des spezifischen Programms kehrt das Systemprogramm wieder zurück an die Unterbrechungsstelle und setzt dort die Bearbeitung fort.

Die Auswertung des Prozessalarms kann pegel- oder flankengetriggert erfolgen. Sie können die Auswertung sperren, verzögern oder freigeben.

Auslösung

Ein Prozessalarm wird durch einen Interrupt ausgelöst.

Anwenderschnittstelle OB 2

Das Systemprogramm ruft bei Auftreten eines Prozessalarms den OB 2 auf. Programmieren Sie hier ein spezifisches Programm, das bei einem Prozessalarm zu bearbeiten ist. Wenn der OB 2 nicht geladen ist, wird die Programmbearbeitung nicht unterbrochen.

Unterbrechungen

Ein Prozessalarmgesteuerte Programmbearbeitung kann nur unterbrochen werden durch:

- einen Programm- oder Gerätefehler (an Befehlsgrenzen)
- Bedienung (PG-Funktion, STOP-Schalter)
- STOP-Befehl.

Hinweis!

Bitte beachten Sie, dass eine alarmgesteuerte nicht durch eine zeitgesteuerte Programmbearbeitung oder eine erneute alarmgesteuerte Programmbearbeitung unterbrochen werden kann!

Mehrfachalarme

Treten während der alarmgesteuerten Programmbearbeitung erneut Prozessalarme auf, werden diese solange ignoriert, bis der OB 2 vollständig bearbeitet ist. Daraufhin kehrt das Systemprogramm wieder zurück zur Unterbrechungsstelle und bearbeitet das Programm bis zur Bausteingrenze. Erst dann wird ein Prozessalarm wieder angenommen, wenn nichts anderes im DX0 definiert ist. Der OB 2 wird auch dann bearbeitet, wenn die Interruptleitung bei Erreichen der Befehlsgrenze bereits wieder passiv ist.

Pegelgetriggerte Prozessalarme

In der Voreinstellung (DX0) ist das Prozessalarmsignal bei der CPU 928 pegelgetriggert, d.h. der aktive Zustand der Interruptleitung setzt eine Anforderung, die an der nächsten Baustein- oder Befehlsgrenze zur Bearbeitung des OB 2 führt.

Ein Prozessalarm wird auch dann erkannt und bearbeitet, wenn das Interrupt-Signal an der Bausteingrenze nicht mehr aktiv ist.

Ein Interrupt, der während des Bearbeitung des OB 2 auftritt, wird nicht erkannt.

Flankengetriggerte Prozessalarme

Die Einstellung erreichen Sie durch die Parametrierung des DX0. Nach der Abarbeitung von OB 2 kann ein neuer Prozessalarm nur durch einen Signalzustandswechsel von inaktiv nach aktiv ausgelöst werden.

Ein Prozessalarm wird auch dann erkannt und bearbeitet, wenn das Interrupt-Signal an der Bausteingrenze nicht mehr aktiv ist.

Sperren der Prozessalarmbearbeitung

Ein alarmgesteuertes Programm wird an einer Bausteingrenze oder einer Befehlsgrenze in das zyklische Programm eingeschoben.

Hier kann es Probleme geben, wenn ein zyklischer Programmteil in einer bestimmten Zeit bearbeitet werden muss.

Sie haben folgende Möglichkeiten eine Alarmunterbrechung zu unterbinden:

- Programmieren Sie diesen Programmteil so, dass er keine Bausteinwechsel enthält und behalten Sie die Voreinstellung im DX0 (Prozessalarme an Bausteingrenzen) bei.
- Verwenden Sie den Befehl "AS" (Prozessalarme sperren) zu Beginn Ihres Programmteils und den Befehl "AF" (Prozessalarm freigeben) am Ende Ihres Programmteils, das nicht unterbrochen werden darf. Zwischen diesen beiden Befehlen erfolgt keine alarmgesteuerte Programmbearbeitung.
- Verwenden Sie die Sonderfunktionen OB 120 und OB 122. Hiermit können Sie die Bearbeitung von auftretenden Prozessalarmen für einen bestimmten Programmteil sperren oder verzögern.

Hinweis!

Die Befehle "AF" und "AS" können nur in Funktionsbausteinen programmiert werden!

Verschachtelte alarm- und zeitgesteuerte Programmbearbeitung

Priorisierung von alarm- und zeitgesteuerter Programmbearbeitung

Wenn während einer zeitgesteuerten Alarmbearbeitung ein Prozessalarm auftritt, wird das Programm an der nächsten Unterbrechungsstelle unterbrochen und der Prozessalarm bearbeitet. Danach wird die zeitgesteuerte Programmbearbeitung zu Ende geführt.

Tritt ein Weckalarm auf, wird zuerst die alarmgesteuerte Programmbearbeitung abgeschlossen und dann die zeitgesteuerte Programmbearbeitung aufgenommen.

Bei gleichzeitigem Auftreten von Weckalarm und Prozessalarm wird zuerst der Prozessalarm und dann der noch anstehende Weckalarm bearbeitet.

Reaktionszeit

Die Reaktionszeit auf eine Weckanforderung entspricht der Bearbeitungszeit eines Bausteins bzw. eines Befehls (je nach gewählter Voreinstellung). Sollten aber zum Zeitpunkt der Unterbrechung der zyklischen Bearbeitung Prozessalarme anstehen, wird das Programm erst dann wieder bearbeitet wenn alle anstehenden Prozessalarme abgearbeitet sind.

Die maximale Reaktionszeit zwischen dem Auftreten und der Bearbeitung eines Weckalarms wächst in diesem Falle um die Bearbeitungszeit der Prozessalarme.

Möchten Sie für einen bestimmten Weckalarm-OB einen Weckfehler weitgehend ausschließen, beachten Sie folgende Regeln:

Es gilt: A+B+C < D.

wobei:
- A = Summe der Bearbeitungszeiten aller höherprioren Programmbearbeitungsebenen (Prozess-, Regler-, Weckalarm-OBs),
- B= Bearbeitungszeit des Weckalarm-OB xy,
- C= Laufzeit des längsten Bausteins aller niederprioren Bearbeitungsebenen,
- D= Zeitraster des Weckalarm-OB xy.

Hinweis!

Wenn Sie Ihr Anwenderprogramm nicht nur zyklisch, sondern auch zeit- bzw. alarmgesteuert bearbeiten lassen, besteht die Gefahr, dass Merker überschrieben werden, wenn die gleichen Merker im zyklischen und eingeschachtelten Programm verwendet werden und das zyklische durch das eingeschachtelte Programm unterbrochen wird.

Merker retten

Mit den OBs 190, 191, 192 und 193 können Sie die Signalzustände ihrer Merker zu Beginn einer zeit- oder alarmgesteuerten Bearbeiten retten und danach wieder zurückschreiben. Näheres hierzu finden sie im Kapitel "Integrierte Sonderfunktionen" .

Teil 6 Unterbrechungs- und Fehlerbehandlung

Überblick

In diesem Kapitel werden die Fehlereinträge beschrieben. Sie erfahren wie Sie Fehler beim Planen und Programmieren Ihrer Programme vermeiden können. Weiterhin werden Sie informiert, welche Hilfen Ihnen das Systemprogramm zur Fehlerdiagnose und evtl. Fehlerreaktion zur Verfügung stellt und in welchen Bausteinen Sie Reaktionen auf bestimmte Fehler programmieren können.

Nachfolgend sind beschrieben:

- Fehlerauswertung
- Steuerbits und USTACK
- Fehler im OB, RUN (DB-Fehler) und im ANLAUF

Inhalt

Thema **Seite**

Teil 6 Unterbrechungs- und Fehlerbehandlung 6-1

Häufige Fehler im Anwenderprogramm 6-2

Fehlerinformationen 6-3

Steuerbits und Unterbrechungsstack 6-7

Fehlerbehandlung über Organisationsbausteine 6-21

Fehler im ANLAUF 6-24

Fehler im RUN und im ANLAUF 6-30

Häufige Fehler im Anwenderprogramm

Das Systemprogramm kann fehlerhaftes Arbeiten der CPU, Fehler in der Systemprogrammbearbeitung oder Auswirkungen einer fehlerhaften Programmierung durch den Anwender feststellen.

Fehlerübersicht

Die folgende Liste enthält eine Aufzählung von Fehlern, die bei der Inbetriebnahme des Anwenderprogramms am häufigsten auftreten, die Sie jedoch schon beim Erstellen Ihres Programmes leicht vermeiden können.

Beachten Sie aus diesem Grund beim Entwurf, bei der Implementierung und bei der Inbetriebnahme Ihres Programms bitte folgende Punkte:

- Bei der Angabe von Byte-Adressen für Ein- und Ausgänge müssen entsprechend adressierte Baugruppen im Zentralgerät oder Erweiterungsgerät stecken.
- Achten Sie darauf, dass alle Operanden mit den korrekten Parametern versorgt werden.
- Vorsicht beim nachträglichen Ändern von Funktionsbausteinen. Kontrollieren Sie, ob die FB/FX mit den richtigen Operanden parametriert sind und ob alle Aktualoperanden angegeben sind.
- Ausgänge, Merker, Zeiten und Zähler sollten nicht an mehreren Stellen im Programm mit entgegengesetzt wirkenden Operationen bearbeitet werden.
- Zeiten sollten nur einmal im Zyklus abgefragt werden (z.B. U T1).
- Sorgen Sie dafür, dass alle im Programm aufgerufenen Datenbausteine vorhanden und ausreichend lang sind.
- Überprüfen Sie, ob alle aufgerufenen Bausteine auch tatsächlich im Speicher vorhanden sind.
- Schmiermerker sollten von alarm- und zeitgesteuerten Programmen gerettet und am Schluss der Bearbeitung wieder geladen werden, wenn diese von anderen Bausteinen (z.B. Standard-FB) benötigt werden.

Fehlerinformationen

Übersicht

Wenn im Anlauf oder bei der zyklischen Bearbeitung des Anwenderprogramms ein Fehler auftritt, stehen Ihnen folgende Fehler-Informationsquellen zur Verfügung:

- LEDs auf der Frontplatte der CPU,
- Unterbrechungsstack USTACK und Steuerbits,
- Systemdatum BS 3, BS 4 und BS 80,
- Fehlerkennungen in AKKU 1 und AKKU 2,
- Bausteinstack BSTACK.

In den nachfolgenden Abschnitten können Sie nachlesen, welche Hilfsmittel es zur Auswertung dieser Informationsquellen gibt und wie Sie die Fehlerinformation für die Analyse einer Störung verwenden können.

LEDs

auf der Frontplatte der CPU

Orientieren Sie sich im Falle eines unerwünschten Stoppzustandes an den Leuchtdioden. Sie können Ihnen Hinweise auf die Fehlerursache geben:

LED-Anzeige	Bedeutung
STOP-LED zeigt Dauerlicht STOP-LED blinkt langsam STOP-LED blinkt schnell	Die unterschiedlichen Anzeigen der STOP-LED deuten auf bestimmte Unterbrechungs- und Fehlerursachen hin. Beachten Sie hierzu die Ausführungen im Teil "Betriebszustände".
ADF-LED zeigt Dauerlicht	Adressierfehler
QVZ-LED zeigt Dauerlicht	Quittungsverzug
ZYK-LED zeigt Dauerlicht	Zykluszeitfehler

Online-Funktion AUSGABE USTACK

Die PG-Online-Funktion AUSGABE USTACK gibt Ihnen Auskunft über die Zustände CPU-interner Steuerbits und den Inhalt des Unterbrechungsstacks (USTACK).

In den USTACK trägt das Systemprogramm beim Übergang in den Stoppzustand alle Informationen ein, die es für einen Wiederanlauf benötigt. Diese Informationen sind:

- Registerinhalte,
- Akkuinhalte,
- STEP-Adresszähler SAZ
- Ergebnisanzeigen.

Diese Einträge sind bei der Fehlerdiagnose eine wertvolle Hilfe.

Vor der Ausgabe des USTACK-Inhalts werden zunächst die Zustände der Steuerbits angezeigt. Diese markieren den aktuellen Betriebszustand, bestimmte Eigenschaften der CPU und des Anwenderprogramms und geben zusätzliche Hinweise auf die Fehlerursache.

Sie können die Funktion "Ausgabe USTACK" in den Betriebszuständen STOP, ANLAUF und RUN benutzen; allerdings erhalten Sie im ANLAUF und im RUN nur Auskunft über die Steuerbits und nicht über den Inhalt des USTACKS.

Die Bedeutung der Steuerbits und der Aufbau des Unterbrechungsstacks sind im folgenden Kapitel beschrieben.

BS 3, BS4

Systemdaten BS 3 und BS 4

Wenn Ihre CPU schon beim ANLAUF infolge eines Fehlers zurück in den Stoppzustand geht, so wird in den Systemdatenwörtern BS3 und BS 4 die Fehlerursache genauer definiert. In diesem Fall handelt es sich um Fehler, auf die das Systemprogramm beim Aufbau der Adresslisten im DB 0 oder bei der Auswertung des DB1, DB 2, DX 0 oder DX 2 stößt.

Die beiden Datenwörter sind unter folgenden absoluten Speicheradressen hinterlegt:

Systemdatenwort BS 3:	KH = EA03
Systemdatenwort BS 4:	KH = EA04

Über die Fehlerkennung im Systemdatenwort BS 3 können Sie feststellen, was für ein Fehler aufgetreten ist.

Das Systemdatenwort BS 4 gibt Ihnen Auskunft darüber, wo der Fehler aufgetreten ist.

Die Fehlerkennungen sind im Datenformat KH eingetragen.

Auswerten von BS 3 und BS 4 mit dem Programmiergerät

Mit der Online-Funktion AUSKUNFT ADRESSE (KH = EA03 bzw. EA04) können Sie den Inhalt der beiden Systemdatenwörter direkt auslesen und so die Fehlerursache ermitteln.

Systemdatum BS 80

Wenn vom Systemprogramm ein schwerer Systemfehler festgestellt wird, so setzt es das Steuerbit INF im Unterbrechungsstack und hinterlegt im Systemdatenwort BS 80 eine zusätzliche Fehlerkennung im Datenformat KH.

Das Systemdatenwort BS 80 hat die absolute Speicheradresse KH = EA50. Sie können es in derselben Weise auslesen wie die Systemdaten BS 3 und BS 4.

Fehlerkennungen

in AKKU 1 und AKKU 2

Treten bei der Programmbearbeitung im ANLAUF oder im ZYKLUS Fehler auf, für die als Anwenderschnittstelle ein bestimmter Organisationsbaustein vorhanden ist, so hinterlegt das Systemprogramm automatisch beim Aufruf des jeweiligen Organisationsbausteins in den Akkumulatoren AKKU 1 und AKKU 2 zusätzliche Fehlerinformationen, die die Fehlerursache näher erläutern (siehe Abschnitt 5.6).

Über die Fehlerkennung im AKKU 1 können Sie feststellen, was für ein Fehler aufgetreten ist.

Eine Fehlerkennung im AKKU 2 (falls eingetragen) gibt Ihnen Auskunft darüber, wo der Fehler aufgetreten ist.

Fehlerkennungen sind im Datenformat KH eingetragen.

Auswerten

von AKKU 1 und AKKU 2 mit dem Programmiergerät

Mit der Online-Funktion AUSGABE USTACK können Sie den Inhalt der beiden Akkumulatoren direkt aus dem USTACK lesen und so die genaue Fehlerursache ermitteln.

von AKKU 1 und AKKU 2 mit STEP® 5 von Siemens

Da die Fehlerkennungen automatisch beim Aufruf eines Fehler-Organisationsbausteins im AKKU 1 und 2 abgelegt werden, können Sie diese Kennungen bei der Programmierung Ihres Fehler-OBs berücksichtigen.

Es ist somit möglich, in einem Organisationsbaustein unterschiedliche Reaktionen auf verschiedene Fehler vorzusehen, in Abhängigkeit von der dort übergebenen Fehlerkennung.

Online-Funktion AUSGABE BSTACK

Die PG-Online-Funktion AUSGABE BSTACK gibt Ihnen Auskunft im STOP über den Inhalt des Bausteinstacks.

Im BSTACK sind, ausgehend vom OB1 bzw. FB0, alle Bausteine aufgeführt, die nacheinander bis zum Übergang in den Stoppzustand aufgerufen und noch nicht zu Ende bearbeitet worden sind. Da der BSTACK von unten gefüllt wird, steht in der obersten Zeile der BSTACK-Ausgabe derjenige Baustein, der als letzter bearbeitet wurde und in dem der Fehler aufgetreten ist.

BSTACK-Informationen

Bei der Auswertung der obersten Zeile erhalten Sie folgende Informationen:

Information	Bedeutung
BAUST.-NR.	Bausteinart und -Nummer des Bausteins, der den fehlerhaften Baustein aufgerufen hat.
BAUST.-ADR.	Absolute Anfangsadresse dieses Bausteins im Anwenderspeicher
RÜCKSPR.-ADR.	Absolutadresse der nächsten zu bearbeitenden Operation dieses Bausteins im Anwenderspeicher
REL.-ADR.	Relativadresse (=Differenz "RÜCKSPR.-ADR."-"BAUST.-ADR.") der nächsten zu bearbeitenden Operation in diesem Baustein (Relativadressen können vom PG in Betriebsart "Eingabesperre"/Schlüsselschalter und mit S5-DOS von Siemens ab Stufe IV über Funktionstaste "Adressen" angezeigt werden.)
DB-NR.	Nummer des in diesem Baustein aufgeschlagenen Datenbausteins
DB-ADR.	absolute Anfangsadresse dieses Datenbausteins (Adresse des Datenwortes DW 0) im Programmspeicher

Beispiel

Ausgabe BSTACK auswerten:

BAUST.-NR.	BAUST.-ADR.	RÜCKSPR.-ADR.	REL.-ADR.	DB.-NR.	DB.-ADR.
OB23	0063	0064	0001	13	0078
FB5	006A	0072	0008	13	0078
FB6	008A	0091	0007	100	0098
OB1	009D	009E	0001		

In diesem Beispiel ist der Stoppzustand im OB23 bei der Bearbeitung derjenigen Anweisung aufgetreten, die im Speicher unter der Absolutadresse "0064 - 1 = 0063" abgelegt ist.

Der OB23 (QVZ-Fehler-OB) ist im FB5 an der relativen Adresse "0008 - 1 = 0007" aufgerufen worden.

Im FB6 ist der Datenbaustein DB 100 aufgeschlagen worden. Beim Übergang der CPU in den Stoppzustand war der Datenbaustein DB 13 gültig.

Der Datenbaustein DB 13 wurde im FB5 aufgeschlagen.

Steuerbits und Unterbrechungsstack

Einleitung

Über die Online-Funktionen AG-INFO und AUSGABE USTACK können Sie mit dem Programmiergerät sowohl Betriebszustand, Eigenschaften der CPU und des Anwenderprogramms als auch eventuelle Fehler- und Unterbrechungsursachen analysieren.

Hinweis!

Die Ausgabe der **Steuerbits** erhalten Sie in **jedem** Betriebszustand, die Ausgabe des **USTACK-Inhalts** nur im **STOP**.

Übersicht

Die Diagnosedaten werden über folgende Bildschirmausgaben dargestellt:

- Steuerbits
  Steuerbits geben den aktuellen bzw. vorausgegangenen Betriebszustand und die Störungsursache an.
  Sind mehrere Fehler aufgetreten, werden in den Steuerbits alle aufgetretenen Fehler dargestellt.

- USTACK
  Im USTACK wird die jeweilige Unterbrechungsstelle (Adressen) mit den dort aktuellen Anzeigen und Akkuinhalten sowie die Störungsursache angegeben.
  Sind mehrere Unterbrechungen aufgetreten, so wird ein mehrstufiger Unterbrechungsstack aufgebaut:
  Tiefe 01 = letzte Unterbrechungsursache,
  Tiefe 02 = vorletzte Unterbrechungsursache usw.
  Bei USTACK-Überlauf (nach mehr als 13 Einträgen) erfolgt ein sofortiger Übergang in den Stoppzustand. Anschließend ist NETZ AUS/NETZ EIN und ein NEUSTART erforderlich.

Die Bedeutung der einzelnen Abkürzungen in den Steuerbits und im Unterbrechungsstack wird nachfolgend erläutert.

Hinweis!

Der Text am Bildschirm Ihres Programmiergerätes ist abhängig von der benutzten PG-Software. Daher kann er von der hier abgedruckten Darstellung abweichen. Trotzdem ist die Beschreibung der Bildschirminformation in dieser Programmieranleitung gültig

Steuerbits

Bei der PG-Ausgabe des USTACK werden auf der 1. Bildschirmseite die Zustände der Steuerbits angezeigt .

STEUERBITS							
>>STP<<	STP-6	FE-STP	BARBEND	PG-STP	STP-SCH	STP-BEF	MP-STP
>>ANL<<	ANL-6	NEUSTAX	M W A	A W A	ANL-2	NEUZUX	MWA-ZULX
>>RUN<< X	RUN-6	EINPROZX	BARB	OB1GEL	FB0GEL X	OBPROZA	OBWECKA
32KWRAM X	16KWRAM	8KWRAM	EPROM	KM-AUS	KM-EIN	DIG-EINX	DIG-AUSX
URGELOE	URL-IA	STP-VER	ANL-ABB	UA-PG	UA-SYS	UA_PRFE	UA-SCH
DX-0-FE	FE-22	MOD-FE	RAM-FE	DB0-FE	DB1-FE	DB2-FE	KOR-FE
N A U	P E U	B A U	STUE-FE	Z Y K	Q V Z	A D F	WECK-FE
B C F	FE-6	FE-5	FE-3	FE-3	L Z F	REG-FE	DOPP-FE

Die Steuerbits (>>STP<<, >>ANL<< und >>RUN<<) und die in den ersten drei Zeilen der 1. Bildschirmseite folgenden Bits markieren den aktuellen bzw. vorausgegangenen Betriebszustand der CPU und geben Auskunft über bestimmte Eigenschaften der CPU und des Anwenderprogramms.

Die Steuerbits lassen sich in allen Betriebszuständen ausgeben. So können Sie sich z.B. jederzeit vergewissern, ob der Organisationsbaustein OB2 geladen und somit eine alarmgesteuerte Programmbearbeitung möglich ist.

Bedeutung der Steuerbits

Zeile >>STP<< (STEUERBITS)	
Steuerbit	**Bedeutung**
>>STP<<	CPU ist im Betriebszustand STOP
STP-6	nicht belegt
FE-STP	Fehler-Stop: Stoppzustand nach NAU (Netzausfall), PEU (Peripherie unklar), BAU (Batterie unklar), STUEB (BSTACK-Überlauf), STUEU (USTACK-Überlauf), DOPP (Doppelfehler) oder CPU-Fehler.
BARBEND	Bearbeitungskontrolle-Ende: Stoppzustand nach Online-Funktion BEARBEITUNGSKONTROLLE ENDE (NEUSTART erforderlich). Wird nicht gesetzt, wenn die Funktion BEARBEITUNGSKONTROLLE ENDE im Stoppzustand der CPU durchgeführt wurde.
PG-STP	PG-STOP: Stoppzustand durch Befehl vom PG
STP-SCH	STOP-Schalter: Stoppzustand durch Betriebsartenschalter in Stellung STOP
STP-BEF	Stopp-Befehl: Stoppzustand nach Bearbeitung der Operation `STP`, Stoppzustand nach Stoppbefehl vom Systemprogramm, wenn Fehler-Organisationsbaustein nicht programmiert ist.

Zeile >>ANL<< (STEUERBITS)	
Steuerbit	**Bedeutung**
>>ANL<<	CPU ist im Betriebszustand ANLAUF
ANL-6 + M W A	MANUELLER NEUSTART
ANL-6 + A W A	AUTOMATISCHER NEUSTART
NEUSTA	MANUELLER NEUSTART ist angefordert (STOP) oder wurde als letzter ANLAUF durchgeführt (ANLAUF/RUN).
M W A	MANUELLER WIEDERANLAUF ist angefordert (STOP) oder wurde als letzter ANLAUF durchgeführt (ANLAUF/RUN).
A W A	AUTOMATISCHER WIEDERANLAUF nach Netzspannungsausfall ist angefordert (STOP) oder wurde als letzter ANLAUF durchgeführt (ANLAUF/RUN).

Fortsetzung ...

.... Fortsetzung

Zeile >>ANL<< (STEUERBITS)	
Steuerbit	**Bedeutung**
M W A + A W A	AUTOMATISCHER NEUSTART wurde angefordert (STOP) oder als letzte Anlaufart wurde AUTOMATISCHER NEUSTART durchgeführt (ANLAUF/ RUN)
ANL-2	Doppelfunktion: wird gesetzt nach Aufruf von BEARBEITUNGSKONTROLLE ENDE (wird im Unterschied zu BARBEND in der ersten Maskenzeile auch dann gesetzt, wenn BEARBEITUNGSKONTROLLE ENDE im STOP aufgerufen wird; verhindert WIEDERANLAUF); wird gesetzt nach "Komprimieren im STOP"; verhindert WIEDERANLAUF.
NEUZU	NEUSTART zulässig (STOP) oder beim letzten ANLAUF war NEUSTART zulässig (ANLAUF/RUN)
MWA-ZUL	MANUELLER WIEDERANLAUF zulässig (STOP) oder beim letzten Anlauf war NEUSTART zulässig (ANLAUF/RUN)

Zeile >>RUN<< (STEUERBITS)	
Steuerbit	**Bedeutung**
>>RUN<<	CPU ist im Zustand RUN (zykl. Bearbeitung aktiv)
RUN-6	nicht belegt
EINPROZ	Einzelprozessorbetrieb
BARB	Online-Funktion BEARBEITUNGSKONTROLLE aktiv
OB1GEL	Organisationsbaustein OB1 ist in den Anwenderspeicher geladen worden. Die zyklische Programmbearbeitung wird durch den OB1 bestimmt.
FB0GEL	Funktionsbaustein FB0 ist in den Anwenderspeicher geladen worden. Die zyklische Programmbearbeitung wird durch den FB0 bestimmt, wenn kein OB1 geladen ist. Wenn FB0 und OB1 geladen sind, gilt der OB1 für die zyklische Programmbearbeitung.
OBPROZA	Prozessalarm-Organisationsbaustein OB2 geladen, d.h. prozessalarmgesteuerte Programmbearbeitung ist möglich.
OBWECKA	Weckalarm-Organisationsbaustein geladen, d.h. zeitgesteuerte Programmbearbeitung ist möglich.

Fortsetzung ...

... Fortsetzung

Zeilen 4 und 5 (STEUERBITS)	
Steuerbit	**Bedeutung**
32 KWRAM	Baugruppe ist im RAM-Betrieb (32 x 2^{10} Wörter)
16 KWRAM	bei Version -3UB21 ohne Bedeutung
8KWRAM	bei Version -3UB21 ohne Bedeutung
EPROM	Baugruppe ist im EPROM-Betrieb (32 x 2^{10} Wörter)
DIG-EIN	Adressliste für digitale Eingänge vorhanden.
DIG-AUS	Adressliste für digitale Ausgänge vorhanden.
URGELOE	CPU wurde urgelöscht (NEUSTART erforderlich).
URL-IA	CPU wird momentan urgelöscht.
STP-VER	CPU hat Stoppzustand des AG verursacht.
ANL-ABB	ANLAUF wurde abgebrochen (NEUSTART erforderlich).
UA-PG	PG hat URLÖSCHEN angefordert.
UA-SYS	Systemprogramm hat URLÖSCHEN angefordert (kein ANLAUF möglich); URLÖSCHEN muss durchgeführt werden.
UA-PRFE	Urlöschanforderung wegen CPU-Fehler
UA-SCH	Urlöschvoranforderung durch Schalterbedienung: URLÖSCHEN durchführen bzw. Wahl einer Anlaufart, wenn angefordertes URLÖSCHEN nicht durchgeführt werden soll.

Die Steuerbits der Zeilen 6 bis 8 (siehe nachfolgende Tabelle) markieren Fehler, die in den Betriebszuständen ANLAUF (z.B. beim ersten NEU-START) und RUN (z.B. bei der zeitgesteuerten Programmbearbeitung) auftreten können.

Sind mehrere Fehler aufgetreten, werden in den letzten drei Zeilen der Steuerbits alle bisher aufgetretenen (und noch nicht bearbeiteten!) Unterbrechungsursachen angezeigt. Beachten Sie hierzu das Systemdatum **BS 2**: Es enthält das UAMK (Unterbrechungsanzeigen-Sammelwort, 16 Bit), in dem ebenfalls alle aufgetretenen und noch nicht bearbeiteten Fehler eingetragen sind (siehe "Speicherbelegung und -organisation").

Bedeutung der Steuerbits

Zeilen 6 bis 8 (STEUERBITS)	
Steuerbit	**Bedeutung**
DX0-FE	Parametrierfehler im DX 0 oder DX 2
FE-22	nicht belegt
MOD-FE	Inhalt der Memory Card ist fehlerhaft (URLÖSCHEN erforderlich).
RAM-FE	Inhalt des Anwenderspeichers oder des DB-RAMs ist fehlerhaft (URLÖSCHEN erforderlich).
DB0-FE	Aufbau der Baustein-Adresslisten im DB 0 ist fehlerhaft.
DB1-FE	Aufbau der Adresslisten in DB 1 für Prozessabbild-Aktualisierung ist fehlerhaft: DB 1 bei Koordinator nicht programmiert, Aufbau oder Inhalt des DB 1 ist fehlerhaft.
DB2-FE	Fehler bei der Auswertung des Parametrierungs-Datenbausteins DB 2 der Reglerstruktur R64
KOR-FE	Fehler bei Datenaustausch mit dem Koordinator (zur Zeit kein Mehrprozessor-Betrieb möglich)
N A U	Netzspannungsausfall im Zentralgerät
P E U	Peripherie unklar = Spannungsausfall im Erweiterungsgerät
B A U	Batterie ist fehlerhaft = Ausfall der Pufferbatterie im Zentralgerät
STUE-FE	Unterbrechungs- oder Bausteinstack übergelaufen (Schachtelungstiefe zu groß; NEUSTART erforderlich)
Z Y K	Zyklusüberwachungszeit überschritten
Q V Z	Quittungsverzug beim Datenaustausch mit Peripherie
A D F	Adressierfehler bei Eingängen oder Ausgängen: Fehler hervorgerufen durch Zugriff auf das Prozessabbild, wobei Peripheriebaugruppen angesprochen wurden, die beim letzten NEUSTART nicht gesteckt, defekt oder nicht im DB 1 angegeben waren.
WECK-FE	Weckfehler: Vor oder während der Bearbeitung eines bestimmten Weckalarm-OBs ist ein weiterer Weckalarm für diesen OB ausgelöst worden.
B C F	Befehlscodefehler: Substitutionsfehler: Bearbeiteter Befehl ist nicht substituierbar, Operationscodefehler: Bearbeiteter Befehl ist falsch, Parameterfehler: Parameter des bearbeiteten Befehls ist falsch.
FE-6	nicht belegt
FE-5	Hinweis auf schweren Systemfehler, Zusatzinformation in BS 80

Zeilen 6 bis 8 (STEUERBITS)	
Steuerbit	Bedeutung
FE-4	Power-down-Fehler: Bearbeitung eines vorausgegangenen Netzausfalls (NAU) durch das Systemprogramm ist fehlerhaft abgelaufen; der WIEDERANLAUF ist deswegen gesperrt.
FE-3	Schnittstellenfehler (SSF)
L Z F	Laufzeitfehler: Aufgerufener Baustein nicht geladen, Lade-/Transferfehler bei Datenbausteinen, Sonstige Laufzeitfehler.
REG-FE	Fehler bei der Bearbeitung der Reglerstruktur R64 im ZYKLUS
DOP-FE	Doppelfehler: Eine noch aktive Fehlerprogrammbearbeitungsebene (ADF, BCF, LZF, QVZ, REG, ZYK) wird ein zweites Mal aktiviert (NEUSTART erforderlich).

USTACK-Inhalt

Befindet sich die CPU im Stoppzustand, so erscheint nach Ausgabe der Steuerbits und Betätigen der Übernahmetaste auf dem Bildschirm des PGs der Inhalt des USTACKS: Das Sytemprogramm trägt beim Übergang in den Stoppzustand alle Informationen in den USTACK ein, die es für einen Wiederanlauf benötigt.

Aus den USTACK-Informationen können Sie bei der Fehlerdiagnose entnehmen, was für ein Fehler aufgetreten ist und an welcher Stelle im Programm er sich ausgewirkt hat.

Hat ein einziger Fehler den Stoppzustand verursacht, so wird nur eine Ebene mit USTACK-Informationen ausgegeben. Sind mehrere Fehler aufgetreten, so werden entsprechend viele Ebenen mit USTACK-Informationen angezeigt (Tiefe 01, Tiefe 02 usw.). In allen Ebenen ist unter STOERUNGSURSACHE ein einziger Fehler angekreuzt.

Bei mehreren Fehlern ist in der Ebene TIEFE 01 der Fehler markiert, der unmittelbar vor dem Übergang in den Stoppzustand festgestellt wurde.

Darstellung am Bildschirm

Ein Beispiel für die PG-Ausgabe des USTACK-Inhalts sehen Sie in folgender Tabelle.

UNTERBRECHUNGSSTACK											
TIEFE	02										
BEF-REG:	C70A	SAZ:	00F3	DB-ADR.:	0000	BA-ADR.:	0000				
BST-STP:	0002	FB-Nr.:	226	DB-Nr.:		OB-Nr.:					
		REL-SAZ:	0006	DBL-REG:	0000						
EBENE:	0004	UAMK:	0200	UALW:	0000						
AKKU1:	0000 C464	AKKU2:	0000 00FF	AKKU3:	0000 0000	AKKU4:	0000 0000				
KLAMMERN:	KE1 111	KE2	100	KE3	111						
ERGEBNISANZEIGE:		ANZ1		ANZ0		OVFL		OVFLS		ODER	ERAB
				X							
		STATUS		VKE							
		X		X							
STOERUNGSURSACHE:		NAU		PEU		BAU		MPSTP		ZYK	QVZ
		ADF		STP		BCF		S-6		LZF	REG-FE
		X									
		STUEB		STUEU		WECK		DOPP			

Erläuterung der USTACK-Anzeigen

Tiefe

Stufe der Informationsebene des USTACK-Inhalts bei Fehlerschachtelung:

TIEFE 01 = zuletzt aufgetretene Störungsursache,

TIEFE 02 = vorletzte aufgetretene Störungsursache,

. . . .

TIEFE 13 = ... (maximale Tiefe)

Angaben über die Fehlerstelle:

Die folgende Tabelle enthält Angaben (USTACK-Kennungen) über die Fehlerstelle, mit denen im Anwenderprogramm die Anweisung gefunden werden kann, bei deren Bearbeitung die CPU in STOP gegangen ist.

Angaben zur Fehlerstelle	
USTACK-Kennung	**Bedeutung**
BEF-REG	Befehlsregister: Es enthält den Maschinencode (1.Wort) des zuletzt bearbeiteten Befehls einer unterbrochenen Programmbearbeitungsebene (siehe Operationsliste, Auflistung des Maschinencodes).
BST-STP	Bausteinstack-Pointer: Er enthält die Anzahl der im Bausteinstack (BSTACK) eingetragenen Elemente zum Zeitpunkt der Unterbrechung dieser Programm-bearbeitungsebene.
EBENE Z	Gibt die Ebene der Programmbearbeitung an, die unterbrochen worden ist:
	Z: 0002: NEUSTART 0004: ZYKLUS 0006: WECKAL./5 s (OB18) 0008: WECKAL./2 s (OB17) 000A: WECKAL./1 s (OB16) 000C: WECKAL./500 ms (OB15) 000E: WECKAL./200 ms (OB14) 0010: WECKAL./100 ms (OB13) 0012: WECKAL./50 ms (OB12) 0014: WECKAL./20 ms (OB11) 0016: WECKAL./10 ms (OB10) 0018: ZEITAUFTRAG 001A: nicht belegt 001C: REGLERALARM 001E: nicht belegt 0020: VERZÖGERUNGSALARM 0022: nicht belegt 0024: PROZESSALARM 0026: nicht belegt 0028: MANUELLER NEUSTART 002A: AUTOMATISCHER NEUSTART 002C: Stoppschalter oder PG-STOP 002E: Schnittstellenfehler

Fortsetzung ...

... Fortsetzung

Angaben zur Fehlerstelle	
USTACK-Kennung	**Bedeutung**
	0030: Weckfehler 0032: Reglerfehler 0034: Zyklusfehler 0036: nicht belegt
	0038: Befehlscodefehler 003A: Laufzeitfehler 003C: Adressierfehler 003E: Quittungsverzug 0040: nicht belegt 0042: nicht belegt 0044: MANUELLER WIEDERANLAUF 0046: AUTOMATISCHER WIEDERANLAUF
SAZ	STEP-Adresszähler: Enthält die Absolutadresse des zuletzt bearbeiteten Befehls einer unterbrochenen Programmbearbeitungsebene im Programmspeicher. Bei Fehler zeigt der SAZ genau auf den fehlerverursachenden Befehl! Vor Ausführung des ersten Befehls einer Bearbeitungsebene steht der SAZ auf ’0’.
...NR.	Bausteinart und -nummer des zuletzt bearbeiteten Bausteins
REL-SAZ	Relativer STEP-Adresszähler: Enthält die Relativadresse (bezogen auf die Bausteinanfangsadresse) des zuletzt bearbeiteten Befehls im zuletzt bearbeiteten Baustein (Relativadressen können vom PG in Betriebsart "Eingabesperre"/ Schlüsselschalter oder mit S5-DOS von Siemens ab Stufe IV über Funktionstaste angezeigt werden oder bei der Ausgabe des Bausteins auf den Drucker).
UAMK	Unterbrechungsanzeigen-Sammelwort: Im UAMK sind alle bisher aufgetretenen und noch nicht zu Ende bearbeiteten Unterbrechungsursachen angezeigt (siehe Teil 8).
UALW	Unterbrechungsanzeigen-Löschwort (siehe Teil "Speicherbelegung und -organisation).
DB-ADR	absolute Anfangsadresse (DW 0) des zuletzt aufgeschlagenen Datenbausteins im Programmspeicher (=0000, wenn kein Datenbaustein aufgeschlagen wurde).
DB-NR	Nummer des zuletzt aufgeschlagenen Datenbausteins.
DBL-REG	Länge des zuletzt aufgeschlagenen Datenbausteins.
BA-ADR	Absolutadresse im Programmspeicher für den nächsten zu bearbeitenden Befehl im zuletzt aufrufenden Baustein.
...NR	Bausteinart und -nummer des zuletzt aufrufenden Bausteins.
AKKU1 bis AKKU4	Inhalt der Rechenregister zum Unterbrechungszeitpunkt: In bestimmten Fehlerfällen werden vom Systemprogramm in AKKU 1 und AKKU 2 Fehlerkennungen hinterlegt, die die Unterbrechungsursache näher erläutern.
KLAMMERN	Anzahl der Klammerebenen: "KEx abc" mit x = 1 bis 7 Ebenen, a = ‘OR’ (ODER, siehe Bitanzeigen), b = ‘VKE’ (Verknüpfungsergebnis, siehe Bitanzeigen), c = 1: ‘U(‘, c = 0: ‘O(‘.

Ergebnisanzeige

siehe Teil "Programmbearbeitung"

Störungsursache

Die folgenden Abkürzungen (USTACK-Kennungen) stellen die wichtigsten Störungsursachen dar.

Es sind nur diejenigen Unterbrechungsursachen angekreuzt, die in der gerade angezeigten Programmbearbeitungsebene (siehe EBENE) aufgetreten sind.

Bei den Angaben der Störungsursachen handelt es sich um die Wiedergabe des Unterbrechungsanzeigensammelwortes (UAMK, 16 Bit; siehe Teil "Speicherbelegung"). Teilweise sind die Angaben hier mit denen der Steuerbits identisch.

Ustack
Störungsursache

STOERUNGSURSACHE	
USTACK-Kennung	**Bedeutung (aufgerufener Fehler-OB)**
N A U	Netzspannungsausfall im Zentralgerät
P E U	Peripherie unklar = Spannungsausfall im Erweiterungsgerät
B A U	Batterie unklar = Ausfall der Pufferbatterie (Zentralgerät)
MPSTP	Wahlschalter am KOR in Stellung STOP. (Zur Zeit wird der Mehrprozessorbetrieb nicht unterstützt.)
ZYK	Zyklusüberwachungszeit überschritten
QVZ	Quittungsverzug beim Datenaustausch mit der Peripherie
ADF	Adressierfehler bei digitalen Eingängen und Ausgängen mit Prozessabbild

Fortsetzung...

... Fortsetzung

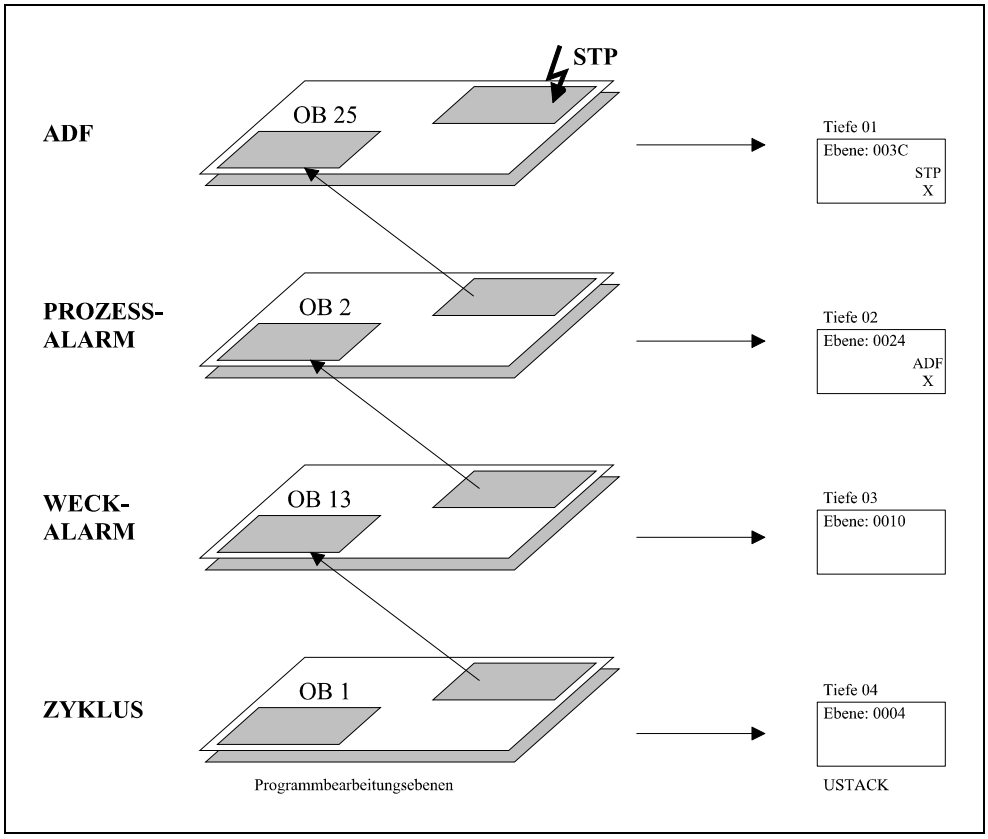
STOERUNGSURSACHE	
USTACK-Kennung	**Bedeutung (aufgerufener Fehler-OB)**
STP	Stoppzustand durch Stoppschalter in Stellung STOP, Stoppzustand durch Befehl vom PG, Stoppzustand nach Bearbeitung der Operation STP, Stoppzustand nach Stoppbefehl vom Systemprogramm, wenn Fehler-Organisationsbaustein nicht programmiert ist.
BCF	Befehlscodefehler: Fehler, die während der Befehlsdecodierung erkannt werden: Substitutionsfehler: Bearbeiteter Befehl ist nicht substituierbar, Operationscodefehler: Bearbeiteter Befehl ist falsch, Parameterfehler: Parameter des bearbeiteten Befehls ist unzulässig.
S-6	Schnittstellenfehler
LZF	Laufzeitfehler: Fehler, die während der Befehlsausführung erkannt werden: aufgerufener Baustein nicht geladen, Lade-Transferfehler bei Datenbausteinen, sonstige Laufzeitfehler.
REG-FE	Fehler bei der Bearbeitung der Reglerstruktur R64 im ZYKLUS
STUEB	Der Bausteinstack ist übergelaufen: Die Schachtelungstiefe ist zu groß; erforderliche Maßnahme: NEUSTART
STUEU	Der Unterbrechungstack ist übergelaufen: Die Schachtelungstiefe ist zu groß; erforderliche Maßnahmen: NEUSTART
WECK	Weckfehler: Vor oder während der Bearbeitung eines bestimmten Weckalarm-OBs ist ein weiterer Weckalarm für diesen OB ausgelöst worden.
DOPP	Doppelfehler: Eine noch aktive Fehlerprogrammbearbeitungsebene (ADF, BCF, LZF, QVZ, REG-FE, ZYK) wird ein zweites Mal aktiviert (NEUSTART erforderlich).

Beispiele zur Fehlerdiagnose über USTACK

Beispiel 1:

Folgende Abbildung zeigt Ihnen den Aufbau des USTACK im Zusammenhang mit den aufgetretenen Unterbrechungen.

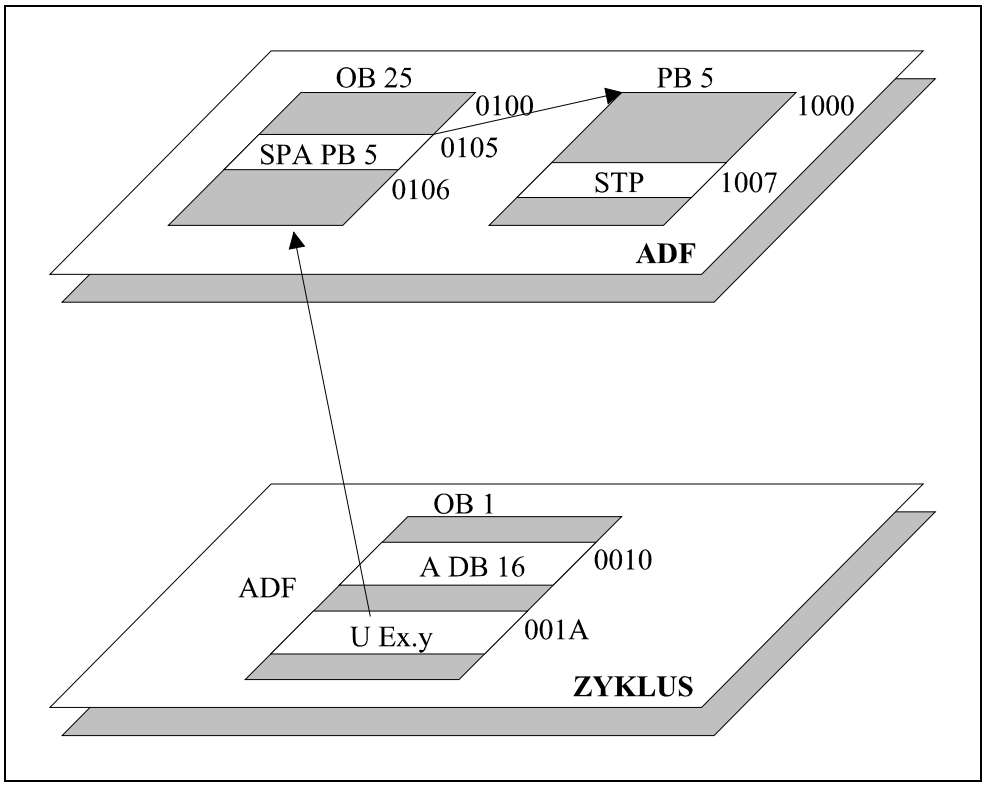
- Die Programmbearbeitungsebene ZYKLUS **(OB1)** wird unterbrochen durch das Auftreten eines Interrupts.
- Daraufhin wird die Programmbearbeitungsebene WECKALARM aktiviert und der **OB13** bearbeitet.
- Durch das Auftreten eines Prozessalarms wird die Ebene WECKALARM verlassen, die Ebene PROZESSALARM aktiviert und der **OB2** bearbeitet.
- Ein falscher Adressierbefehl führt dazu, dass die Ebene ADF aktiviert und dort der **OB25** bearbeitet wird. In seinem Fehlerbehandlungsprogramm hat der Anwender einen Stoppbefehl (STP) programmiert: Die CPU bricht die Programmbearbeitung ab.



Vor dem endgültigen Übergang in den Stoppzustand sind insgesamt vier verschiedene Programmbearbeitungsebenen unterbrochen worden. Wenn Sie sich nun am PG den USTACK ausgeben lassen, bekommen Sie einen vierstufigen USTACK, zuoberst der USTACK mit der TIEFE 01, in dem die Kennung der zuletzt unterbrochenen Programmbearbeitungsebene (=ADF) vermerkt ist. Sie können den USTACK nun "hinunterschalten" bis zum USTACK mit der Tiefe 04, der die Programmbearbeitungsebene ZYKLUS repräsentiert, die als erste unterbrochen wurde.

Beispiel 2:

In diesem Beispiel erkennt die CPU bei der Ausführung der Operation U Ex.y im OB1 einen Adressierfehler. Dies führt zur Bearbeitung des OB25. Aufgrund einer STP-Operation im PB 5 geht die CPU in den Stoppzustand.



Fortsetzung...

... Fortsetzung Zwei unterbrochene Programmbearbeitungsebenen führen zum Aufbau eines zweistufigen USTACKS:

UNTERBRECHUNGSSTACK							
TIEFE	01						
BEF-REG:	STP	SAZ:	1007	DB-ADR.:		BA-ADR.:	0106
BST-STP:	0003	PB-Nr.:	5	DB-Nr.:	16	OB-Nr.:	26
		REL-SAZ:	0007	DBL-REG:			
EBENE:	003C	UAMK:	0300	UALW:	0000		
AKKU1:							
ERGEBNISANZEIGE:		. . .					
STOERUNGSURSACHE:							
		STP					
		X					

UNTERBRECHUNGSSTACK							
TIEFE	02						
BEF-REG:	U Ex.y	SAZ:	001A	DB-ADR.:		BA-ADR.:	
BST-STP:	0001	OB-Nr.:	1	DB-Nr.:	16	OB-Nr.:	
		REL-SAZ:	000A	DBL-REG:			
EBENE:	0004	UAMK:	0200	UALW:	0000		
AKKU1:							
ERGEBNISANZEIGE:		. . .					
STOERUNGSURSACHE:							
		ADF					
		X					

Fehlerbehandlung über Organisationsbausteine

Einleitung

Wenn das Systemprogramm einen bestimmten Fehler erkannt hat, ruft es den für diesen Fall vorgesehenen Organisationsbaustein auf.

Durch entsprechende Programmierung dieses Organisationsbausteins können Sie nun das weitere Verhalten der CPU festlegen.

Abhängig davon, wie Sie den Organisationsbaustein programmieren, können Sie:

- die normale Programmbearbeitung fortsetzen lassen,
- die CPU in STOP bringen,
  und / oder
- ein spezielles "Fehlerprogramm" bearbeiten lassen.

Bei Fehlern aufgerufene OBs

Fehlerursache	Aufruf von	Reaktionen bei nicht geladenem OB [1)]
Aufruf eines nicht geladenen Bausteins (LZF)	OB19	STOP
Quittungsverzug im Anwenderprogramm bei Zugriff auf Peripheriebaugruppen (QZV)	OB23	keine
Quittungsverzug beim Aktualisieren des Prozessabbildes	OB24	keine
Adressierfehler (ADF)	OB25	STOP
Zykluszeitüberschreitung (ZYK)	OB26	STOP
Substitutionsfehler (BCF)	OB27	STOP
Betriebsartenschalter auf STOP, PG-Funktion AG-STOP	OB28	STOP
Operationscodefehler (BCF)	OB29	STOP
Parameterfehler (BCF)	OB30	STOP
Sonstige Laufzeitfehler (LZF)	OB31	STOP
Lade-/Transferfehler bei Datenbausteinen (TRAF)	OB32	STOP
Weckfehler (WECK-FE)	OB33	STOP
Fehler bei der Bearbeitung der Reglerstruktur R64 (REG-FE)	OB34	STOP
Kommunikationsfehler auf der zweiten seriellen Schnittstellen (FE-3)	OB35	keine

[1)] DX-0-Voreinstellung

Reaktion bei nicht geladenem OBs

Die Reaktion bei nicht geladenem Organisationsbaustein ist fehlerabhängig:

Keine Unterbrechung der zyklischen Programmbearbeitung

Tritt ein Quittungsverzug auf und OB23, OB24 oder OB35 sind nicht geladen, so wird die zyklische Programmbearbeitung nicht unterbrochen.

Es erfolgt keine Reaktion der CPU.

Soll die CPU bei QVZ in den Stoppzustand übergehen, so muss der Organisationsbaustein eine Stopp-Anweisung enthalten und mit BE abgeschlossen oder der DX 0 entsprechend parametriert werden.

Programm für STOP:

```
:
:
:STP
:BE
```

Stoppzustand

In allen übrigen Fehlerfällen geht die CPU sofort in den Stoppzustand, wenn die zugehörigen Organisationsbausteine vom Anwender nicht programmiert worden sind.

Soll in Ausnahmefällen (z.B. während der Inbetriebsetzung) der eine oder andere Fehler die zyklische Programmbearbeitung nicht unterbrechen, so genügt eine Bausteinende-Anweisung im jeweiligen Organisationsbaustein oder eine entsprechende Parametrierung des DX 0.

Programm für Betrieb ohne Unterbrechung:

```
:
:
:BE
```

Hinweis!

Eine Ausnahme bildet der Organisationsbaustein OB28: Hier erfolgt immer ein Übergang in den Stoppzustand, unabhängig davon, ob der OB28 geladen ist und wie er programmiert ist.

Wenn Sie den betreffenden Organisationsbaustein nicht programmieren wollen, haben Sie die Möglichkeit, durch entsprechende Parametrierung des Datenbausteins DX 0 den Übergang der CPU in den Stoppzustand zu verhindern.

Unterbrechungen bei der Bearbeitung der Fehler-OBs

Nachdem das Systemprogramm den betreffenden Organisationsbaustein aufgerufen hat, wird das darin enthaltene Anwenderprogramm bearbeitet. Tritt während der Bearbeitung eines Organisationsbausteins erneut ein Fehler auf, wird wie in der zyklischen Programmbearbeitung das Programm an der nächsten Befehlsgrenze unterbrochen und der entsprechend andere Organisationsbaustein aufgerufen.
Die Organisationsbausteine werden in der Reihenfolge bearbeitet, in der sie aufgerufen werden. Wieviele Fehler-Organisationsbausteine ineinandergeschachtelt werden können, ist abhängig von:

- Art der Fehler
- Anzahl der Programmbearbeitungsebenen im Fehlerfall
- Anzahl der aufgerufenen Bausteine im Fehlerfall

Art der aufgetretenen Fehler:

Es können keine Organisationsbausteine ineinander geschachtelt werden, die derselben Programmbearbeitungsebene angehören. (Zur Zuordnung der Fehler-OB zu den Programmbearbeitungsebenen siehe folgendes Kapitel).

Bei der Bearbeitung des OB27 (Programmbearbeitungsebene BCF) kann beispielsweise ein OB32 (Programmbearbeitungsebene LZF), nicht jedoch ein OB29 oder OB30 (ebenfalls BCF) eingeschachtelt werden.
Bei Doppelaufruf einer Programmbearbeitungsebene geht die CPU unmittelbar in STOP.

Anzahl der zu diesem Zeitpunkt aktivierten Programmbearbeitungsebenen:

Für jede aktivierte Programmbearbeitungsebene benötigt das Systemprogramm bei Unterbrechungen besonderen Speicherplatz zum Anlegen des USTACKs. Reicht dieser Speicherplatz nicht mehr aus, so kommt es zu einem USTACK-Überlauf.

Bei USTACK-Überlauf geht die CPU unmittelbar in STOP:

Anzahl der zu diesem Zeitpunkt aufgerufenen Bausteine:

Bei BSTACK-Überlauf geht die CPU unmittelbar in STOP.

Fehler im ANLAUF

Einleitung

Bei der Initialisierung und im Anlauf auftretende Störungs- und Fehlerursachen können dazu führen, dass das Anlaufprogramm abgebrochen wird und die CPU in den Stoppzustand übergeht.

Im Anlaufprogramm (Organisationsbausteine OB20, 21 und 22) auftretende Unterbrechungsursachen werden wie im Zyklus behandelt.

Ausnahme: Bei einem STOP im Anlauf wird kein Organisationsbaustein OB28 aufgerufen.

Unterbrechungs- und Fehlerursachen im Anlauf

Auf die in der folgenden Tabelle genannten Unterbrechungs- und Fehlerursachen gibt es **keine Reaktionsmöglichkeit** durch eine Anwenderschnittstelle (Fehler-OB).

Steuerbit oder Kennung im USTACK	Erläuterung
STP	Stoppbefehl vom Systemprogramm (bei FE-STP) oder im Anwenderprogramm
BAU	Ausfall der Pufferbatterie im Zentralgerät
NAU	Ausfall der Versorgungsspannung im Zentralgerät
PEU	Ausfall der Versorgungsspannung in einem Erweiterungsgerät
STUEU	Stacküberlauf beim Unterbrechungsstack (USTACK)
STUEB	Stacküberlauf beim Bausteinstack (BSTACK)
DOPP-FE	Doppelaufruf einer Fehlerprogrammbearbeitungsebene
RAM-FE	Fehler bei der Initialisierung: Inhalt des Betriebssystem-RAMs oder des DB-RAMs defekt
MOD-FE	Fehler bei der Initialisierung: Inhalt der Memory Card nicht korrekt
DB0-FE [1)]	Fehler beim Aufbau der Bausteinadressliste (DB0)
DB1-FE [1)]	Fehler bei der Auswertung des DB 1 zum Aufbau der Adressliste für die Prozessabbildaktualisierung
DB2-FE [1)]	Fehler bei der Auswertung des DB 2 der Reglerstruktur R64
DX0-FE [1)]	Fehler bei der Auswertung des Datenbausteins DX 0 oder Fehler bei der Auswertung des Datenbausteins DX 2

[1)] Diese Fehler werden nachfolgend näher erläutert.

DB0-FE (DB-0-Fehler)

Fehler beim Aufbau der Bausteinadressliste (Datenbaustein DB 0).

Der DB 0 wird vom Systemprogramm nach URLÖSCHEN aufgebaut. Bei einem DB-0-Fehler finden Sie in den Systemdatenwörtern BS 3 und BS 4 Fehlerkennungen, die den aufgetretenen Fehler näher definieren.

Kennungen für DB-0-Fehler

Fehlerkennung BS 3	BS 4	Erläuterung
8001h	yyyyh	Falsche Bausteinlänge yyyy = Adresse des Bausteins mit falscher Länge
8002h	yyyyh	Berechnete Endadresse des Bausteins im Speicher falsch yyyy = Bausteinadresse
8003h	yyyyh	Ungültige Bausteinkennung yyyy = Adresse des Bausteins mit falscher Kennung
8004h	yyyyh	Zu große Organisationsbausteinnummer (erlaubt: OB 1 bis OB 39) yyyy = Adresse des Bausteins mit falscher Nummer
8005h	yyyyh	Datenbausteinnummer 0 (erlaubt: DB 1 bis DB 255) yyyy = Adresse des Bausteins mit falscher Nummer

DB1-FE (DB-1-Fehler)

Fehler bei der Auswertung des DB 1 zum Aufbau der Adressliste für die Prozessabbildaktualisierung

- fehlerhafte DB-1-Adressliste bei NEUSTART.

Hinweis!
Die Auswertung der DB-1-Parameter erfolgt jedoch nur bei NEUSTART!

Kennungen für DB-1-Fehler

Fehlerkennung BS 3	BS 4	Erläuterung
0410h	yyyyh	Unzulässige Kennung: Kopfkennung fehlt oder fehlerhaft (korrekt KC 'MASK01') Kennung unzulässig (zulässig KH DE00, DA00, CE00, CA00, BB00) Endekennung fehlt oder fehlerhaft (korrekt KH EEEE) yyyy = unzulässige Kennung
0411h	yyyyh	"Digitale Eingänge", Anzahl Adressen unzulässig (zulässig: 0...128) yyyy = unzulässige Anzahl Adressen
0412h	yyyyh	"Digitale Ausgänge", Anzahl Adressen unzulässig (zulässig: 0...128) yyyy = unzulässige Anzahl Adressen
0415h	yyyyh	Ungültige Anzahl Zeitzellen (zulässig: 0...256) yyyy = unzulässige Anzahl Zeitzellen
0419h	yyyyh	Quittungsverzug bei digitalen Eingängen yyyy = Adresse des nicht quittierten Eingangsbytes
041Ah	yyyyh	Quittungsverzug bei digitalen Ausgängen yyyy = Adresse des nicht quittierten Ausgangsbytes
041Bh	yyyyh	Ouittungsverzug bei Koppelmerker-Eingang yyyy = Adresse des nicht quittierten Koppelmerkerbytes
041Ch	yyyyh	Quittungsverzug bei Koppelmerker-Ausgang yyyy = Adresse des nicht quittierten Koppelmerkerbytes

DX0-FE (DX-0- oder DX-2-Fehler)

Hinweis!
DX-0-Fehler und DX-2-Fehler haben ein gemeinsames Steuerbit (DX0-FE) in der Steuerbit-Maske.

Fehler bei der Auswertung des Datenbausteins DX 0
Bei einem DX-0-Fehler finden Sie in den Systemdatenwörtern BS 3 und BS 4 Fehlerkennungen, die den aufgetretenen Fehler näher definieren.

Kennungen für DX-0-Fehler

Fehlerkennung BS 3	BS 4	Erläuterung
0431h	yyyyh	Unzulässige Kennung: Kopfkennung fehlt oder fehlerhaft (korrekt KC `MASKX0') Blockkennung unzulässig Endekennung fehlt oder fehlerhaft (korrekt KH EEEE) yyyy = unzulässige Kennung
0432h	yyyyh	Unzulässiger Parameter yyyy = unzulässiger Parameter
0433h	yyyyh	Nicht erlaubte Anzahl Zeitzellen (erlaubt: 0...256) yyyy = falsche Anzahl Zeitzellen
0434h	yyyyh	Unerlaubte Zyklusüberwachungszeit (erlaubt: 1 ms bis 13000ms) yyyy = falsche Zeitgröße

Fehler bei der Auswertung des Datenbausteins DX 2

Parametrierung der 2. seriellen Schnittstelle:

Der Datenbaustein DX 2 wird vom Systemprogramm bei NEUSTART geprüft. Bei einem DX-2-Fehler finden Sie in den Systemdatenwörtern BS 3 und BS 4 Fehlerkennungen, die den aufgetretenen Fehler näher definieren.

Kennungen DX-2-Fehler

Fehlerkennung BS 3	BS 4	Erläuterung
0451h	-	DX-2-Länge (ohne Bausteinkopf) < 4 Wörter unzulässig
0452h	yyyy	DX-2-Länge (ohne Bausteinkopf) ist für Kopplungstyp zu kurz yyyy = Länge DX 2
0453h	yyyy	Kopplungstyp unzulässig yyyy = Kopplungstyp
0454h	xx00h	Datenkennung für statischen Parametersatz ungültig (ungleich 44H, 58H) xx = Datenkennung
0455h	xxyyh	Baustein für statischen Parametersatz unzulässig xx = Kennung / yy = DB-Nummer
0456h	xxyyh	Statischer Parametersatz nicht vorhanden xx = Kennung / yy = DB-Nummer
0457h	yyyyh	Statischer Parametersatz zu kurz yyyy = Nummer des nicht vorhandenen DW
0458h	xx00h	Datenkennung für dyn. Parametersatz ungültig (ungleich 44H, 58H, 00H) xx = Datenkennung
0459h	xxyyh	Baustein für dyn. Parametersatz unzulässig xx = Kennung/ yy = DB-Nummer
045Ah	xx00h	Datenkennung für Sendefach/Auftragsfach ungültig (ungleich 44H, 58H, 00H) xx = Datenkennung
045Bh	xxyyh	Baustein für Sendefach/Auftragsfach unzulässig xx = Kennung / yy = DB-Nummer
045Ch	xx00h	Datenkennung für Empfangsfach ungültig (ungleich 44H, 58H, 00H) xx = Datenkennung
045Dh	xxyyh	Baustein für Empfangsfach unzulässig xx = Kennung / yy = DB-Nummer
045Eh	xx00h	Datenkennung für Koordinierungsbytes ungültig (ungleich 44H, 58H, 4DH) xx = Datenkennung
045Fh	xxyyh	Baustein für Koordinierungsbytes unzulässig xx = Kennung / yy = DB-Nummer
0460h	xxyyh	Baustein für Koordinierungsbytes nicht vorhanden xx = Kennung / yy = DB-Nummer
0461h	yyyyh	Datenwort für Koordinierungsbytes nicht vorhanden yyyy = Nr. des nicht vorhandenen DW

MOD-FE (Memory-Card-Fehler)

Beim Auswerten einer Memory Card und beim Kopieren der Bausteine von der Memory Card werden verschiedene Prüfungen durchgeführt. Im Fehlerfall wird das Steuerbit MOD-FE in der Steuerbit-Maske sowie eine zusätzliche Kennung in das Systemdatenwort BS 3 eingetragen.

Fehlerkennungen in BS 3

Wenn die oben genannten Prüfungen zu einem Fehler führen, finden Sie im Systemdatenwort BS 3 eine Fehlerkennung, die den aufgetretenen Fehler näher definiert (der Inhalt von BS 4 ist irrelevant).

Kennungen für Memory-Card-Fehler und Fehler beim Kopieren der Bausteine

Fehlerkennung BS 3	BS 4	Erläuterung
620Eh	-	Memory Card: Falsche Zugriffszeitklasse
6210h	-	Memory Card: Falsche Datenbreite
6211h	-	Memory Card: Falsche Applikation (nicht STEP® 5 von Siemens)
6212h	-	Memory Card: Falsche MLFB-Nummer
6213h	-	Memory Card: Falsche Klasse (kein Flash)
6214h	-	Ungültiger Baustein
6215h	-	Unzulässige Bausteinnummer
6216h	-	Unzulässiger Bausteintyp
6217h	-	Unzulässige Bausteinlänge
6218h	-	Zu viele Bausteine
6219h	-	Zu viele Bausteine eines Typs
621Ah	-	Kein Platz im Anwenderspeicher
621Bh	-	Inhalt der Memory Card inkonsistent

Fehler im RUN und im ANLAUF

Einleitung

Im Betriebszustand RUN kann eine zyklische, eine zeit- oder alarmgesteuerte Programmbearbeitung oder eine Reglerbearbeitung an Befehlsgrenzen unterbrochen werden durch das Auftreten von Störungen wie z.B. Ausfall der Versorgungsspannung am Zentralgerät oder Überlauf des Bausteinstacks.

Bei der Initialisierung und im Betriebszustand ANLAUF auftretende Unterbrechungsursachen führen ebenfalls dazu, dass das Anlaufprogramm abgebrochen wird und die CPU in den Stoppzustand übergeht bzw. den für diesen Fehlerfall vorgesehenen Organisationsbaustein aufruft. Im Anlaufprogramm auftretende Unterbrechungsursachen werden wie im ZYKLUS behandelt.

Man unterscheidet zwischen Störungen, die die CPU direkt in den Betriebszustand STOP überführen (z.B. STUEU) und Störungen, bei deren Auftreten das Systemprogramm anstelle eines Übergangs in den Stoppzustand bestimmte Organisationsbausteine aufruft (z.B. ADF), die der Anwender programmieren kann.

Auf die in den folgenden beiden Tabellen genannten Unterbrechungs- und Fehlerursachen gibt es **keine Reaktionsmöglichkeit** durch eine Anwenderschnittstelle (Fehler-OB).

Fehler, die direkt in den STOP führen

Bei diesen Fehlern wird ein USTACK aufgebaut, in dem die aufgetretene Störung angezeigt wird.

Steuerbit oder Kennung im USTACK	Erläuterung
STP	STOP durch das Systemprogramm (bei Maschinenfehler), wenn ein Fehler-OB nicht geladen ist, oder Stoppbefehl im Anwenderprogramm
BAU	Ausfall der Pufferbatterie am Zentralgerät
NAU	Ausfall der Versorgungsspannung im Zentralgerät
PEU	Ausfall der Versorgungsspannung in einem Erweiterungsgerät
STUEU	Stacküberlauf beim Unterbrechungsstack bei zu großer Schachtelungstiefe (USTACK)
STUEB	Stacküberlauf beim Bausteinstack bei zu großer Schachtelungstiefe (BSTACK)
DOPP-FE	Doppelaufruf einer Fehlerprogrammbearbeitungsebene

Fehler, bei denen ein Fehler-OB aufgerufen wird

Fehler- und Unterbrechungsursachen im ANLAUF und RUN

Bei diesen Steuerbit oder Kennung im USTACK	Erläuterung	aufgerufener OB
BCF	Befehlscodefehler:	
	Substitutionsfehler	OB27
	Operationscodefehler	OB29
	Parameterfehler	OB30
LZF	Laufzeitfehler:	
	Aufruf eines nicht geladenen Bausteins	OB19
	Transferfehler bei DBs	OB32
	Sonstige Laufzeitfehler	OB31
ADF	Adressierfehler:	
	Bei Zugriff auf Prozessabbild	OB25
QVZ	Quittungsverzug:	
	Im Anwenderprogramm bei Zugriff auf Peripheriebaugruppen	OB23
	Bei der Prozessabbildaktualisierung	OB24
ZYK	Zyklusfehler:	
	Überschreitung der Zyklusüberwachungszeit	OB26
WECK-FE	Weckfehler:	
	Fehler bei Bearbeitung eines Weckalarms	OB33
REG-FE	Reglerfehler:	
	Fehler bei Bearbeitung eines Regleralarms	OB34
ABBR	Abbruch:	
	siehe "ABBR" in diesem Abschnitt	OB28
S-6	Kommunikationsfehler:	
	Bei Datenverkehr über die zweite serielle Schnittstelle	OB35

Auf den nachfolgenden Seiten ist jede dieser Fehlerursachen genauer beschrieben:

BCF (Befehlscode-fehler)

Ein Befehlscodefehler tritt dadurch auf, dass die CPU einen Befehl des Anwenderprogramms nicht interpretieren oder ausführen kann. Alle zulässigen Befehlscodes sind in der Operationsliste aufgelistet.

Der Befehl, der den entsprechenden Befehlscodefehler verursacht, wird nicht ausgeführt. Falls der entsprechende BCF-Organisationsbaustein geladen ist, wird dieser aufgerufen, bearbeitet und anschließend mit dem nächsten Befehl im unterbrochenen Anwenderprogramm fortgefahren. Bei nicht geladenem BCF-OB geht die CPU in den Stoppzustand.

Es werden folgende Befehlscodefehler unterschieden, bei denen jeweils der genannte Fehler-OB aufgerufen wird.

Substitutions-fehler (OB27)

Wenn in einem Funktionsbaustein eine Operation mit einem Formal-operanden ausgeführt werden soll, so ersetzt die CPU bei der Bearbeitung des Anwenderprogramms diesen Formaloperanden durch den im Aufruf des Funktionsbausteins stehenden Aktualoperanden.

Erkennt die CPU eine unzulässige Substitution, so unterbricht das System-programm daraufhin die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein OB27 auf, wenn dieser geladen ist.

AKKU 1 enthält zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

BCF - Substitutionsfehler

Fehlerkennung AKKU-1-L	Fehlerkennung AKKU-2-L	Erläuterung
1801h	-	Substitutionsfehler beim Befehl B BS
1802h	-	Substitutionsfehler bei den Befehlen B DW, B MW
1803h	-	Substitutionsfehler bei den Befehlen B=, BI
1804h	-	Substitutionsfehler bei den Befehlen L=, T=
1805h	-	Substitutionsfehler bei den Befehlen U=, UN=, O=, ON=, ==, S=, RB=
1806h	-	Substitutionsfehler bei den Befehlen RD=, LC=, FR=, SAR=, SE=, SSV=, SVZ=

Operationscode-fehler (OB29)

Ein Operationscodefehler wird von der CPU bei der Bearbeitung eines Programmes festgestellt, wenn ein Befehl programmiert worden ist, der nicht zum Befehlsumfang der CPU 928 gehört. Es können z.B. RU- und SU-Befehle mit dem PG programmiert, jedoch von den CPUs 928 nicht interpretiert werden.
Beim Erkennen eines unzulässigen Operationscodes wird an dieser Stelle die Bearbeitung des Anwenderprogramms unterbrochen und der Organisationsbaustein OB29 aufgerufen, wenn er geladen ist.

Bei Aufruf des OB29 stehen im AKKU 1 zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

BCF - Operationscodefehler

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1811h	-	Befehl mit unzulässigem Opcode
1812h	-	Unzulässiger Opcode bei einem Befehl, bei dem das High-Byte des 1. Befehlswortes den Wert 68h enthält.
1813h	-	Unzulässiger Opcode bei einem Befehl, bei dem das High-Byte des 1. Befehlswortes den Wert 78h enthält.
1814h	-	Unzulässiger Opcode bei einem Befehl, bei dem das High-Byte des 1. Befehlswortes den Wert 70h enthält.
1815h	-	Unzulässiger Opcode bei einem Befehl, bei dem das High-Byte des 1. Befehlswortes den Wert 60h enthält.

Achtung!

Ein Operationscodefehler sollte nicht quittiert werden: Die CPU erkennt nicht, ob es sich bei dem fehlerhaften Befehl um einen Einwort- oder Mehrwortbefehl handelt. Hat die CPU den OB29 bearbeitet, so versucht sie, das Programm mit dem nächsten Befehlswort fortzusetzen. Falls es sich dabei um das zweite Wort eines Mehrwortbefehls handelt, erkennt sie entweder einen weiteren Befehlscodefehler oder führt dieses Wort als gültigen Befehl aus, was zu beliebigen Programmfehlern führen kann.

Parameterfehler (OB30)

Ein unzulässiger Parameter tritt auf, wenn ein Befehl mit einem Parameter, der für die entsprechende CPU unzulässig ist, programmiert worden ist (z.B. Aufruf eines reservierten Datenbausteins) oder wenn eine nicht vorhandene Sonderfunktion aufgerufen wird.

Wenn ein unzulässiger Parameter von der CPU erkannt wird, unterbricht das Systemprogramm die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein OB30 auf, wenn dieser geladen ist.

Beim Aufruf des OB30 stehen im AKKU 1 zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

BCF-Parameterfehler

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1821h	-	A DB 0, 1, 2
182Bh	-	SPA(B) OB0
182Ch	-	SPA(B) OB > 39: Sonderfunktion nicht vorhanden
182Dh	-	AX DX 0, AX DX 1, AX DX 2
182Eh	-	L MW/T MW/L PW/T PW/L QW/T QW/L DD/T DD/ B MW 255
182Fh	-	L EW/T EW/L AW/T AW 127
1830h	-	L MD/T MD 253, 254, 255
1831h	-	L ED/T ED/L AD/T AD 125, 126, 127
1832h	-	RLD/RRD/SVD/SLD 33-255
1833h	-	SLW/SRW/LIR/TIR 16-255
1834h	-	SES/SEF 32-255
1835h	-	U=/UN=/O=/ON=/S=/RB=/==/RD=/FR=/SI=/SE=/ SVZ=/SSV=/SAR=/L=/LC=/LW=/T= 0, 127-255
1836h	-	B=/LD= 0, 126-255
1837h	-	U S/O S/S S/= S/UN S/ON S/R S Bytenummer > 1023
1838h	-	U S/O S/S S/= S/UN S/ON S/R S Bitnummer > 7
1839h	-	L SY/T SY Parameter > 1023
183Ah	-	L SW/T SW Parameter > 1022
183Bh	-	L SD/T SD Parameter > 1020
183Ch	-	E DB/EX DX Parameter 0, 1 oder 2 (DB bzw. DX 0, 1, 2 nicht erzeugbar)

LZF (Laufzeitfehler)

Ein Laufzeitfehler tritt dadurch auf, dass die CPU während der Bearbeitung eines Befehls einen Fehler erkennt.

Der Befehl, der den entsprechenden Laufzeitfehler verursacht, wird nicht ausgeführt. Ist ein LZF-Organisationsbaustein vorhanden, so wird dieser aufgerufen. Anschließend wird das unterbrochene Anwenderprogramm mit dem nächsten Befehl, der dem fehlerverursachenden Befehl folgt, fortgesetzt. Ist kein LZF-OB geladen, so geht die CPU in den STOP.

Es werden folgende Laufzeitfehler unterschieden, bei denen jeweils der genannte Fehler-OB aufgerufen wird.

Aufruf eines nicht geladenen Bausteins (OB19)

Wenn im Anwenderprogramm ein Baustein aufgerufen oder aufgeschlagen wird, der nicht vorhanden ist, erkennt das Systemprogramm einen Fehler. Dies gilt für alle Bausteinarten und sowohl für bedingte als auch für die unbedingte Aufruf-Anweisung.

Wenn der Aufruf oder das Aufschlagen eines nicht geladenen Bausteins erkannt wird, ruft das Systemprogramm den Organisationsbaustein OB19 auf, wenn er geladen ist. Im OB19 können Sie das weitere Verhalten der CPU festlegen.

Falls ein OB19 programmiert ist, wird dieser aufgerufen und anschließend die Bearbeitung des unterbrochenen Programms mit dem nächsten Befehl fortgesetzt außer, wenn der OB19 mit einem Stoppbefehl programmiert ist. Ist hingegen der OB19 nicht programmiert, so geht die CPU beim Aufruf oder Aufschlagen eines nicht geladenen Bausteins in den Stoppzustand.

Bei Aufruf des OB19 stehen im AKKU 1 zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

LZF - Aufruf eines nicht geladenen Bausteins

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A01h	-	Nicht geladener Datenbaustein bei A DB
1A02h	-	Nicht geladener Datenbaustein bei AX DX
1A03h	-	Nicht geladener Baustein bei SPA(B) FB, OB1 bis OB39, PB, SB
1A04h	-	Nicht geladener Baustein bei BA(B) FX
1A05h	-	Nicht geladener Datenbaustein bei OB254 bzw. OB255
1A06h	-	Nicht geladener Datenbaustein bei OB182
1A07h	-	Nicht geladener Datenbaustein bei OB150/OB151/OB153

Hinweis!

Beim Versuch, einen nicht geladenen Datenbaustein aufzuschlagen, wird das DBA-Register beeinflusst. In diesem Fall muss vor Zugriffen auf DB/DX-Daten erneut ein geladener Datenbaustein aufgeschlagen werden.

Lade-/ Transferfehler (OB32)

Beim Transferieren von Daten in Datenbausteine (DB, DX) vergleicht die CPU die Länge des aufgeschlagenen DBs mit dem im Transfer-Befehl stehenden Operanden. Wird durch den angegebenen Parameter die Datenbausteinlänge überschritten, so wird die Transfer-Anweisung nicht ausgeführt, um ein irrtümliches Überschreiben von Daten im Speicher zu verhindern.

Ein Lade-/Transferfehler wird auch festgestellt, wenn ein einzelnes Bit innerhalb eines nicht vorhandenen Datenwortes abgefragt oder verändert werden soll.

Ein Lade-/Transferfehler wird ebenfalls erkannt, wenn ein Zugriff auf ein Datenwort stattfinden soll, bevor ein Datenbaustein aufgeschlagen ist (mit A DBn bzw. AX DXn).

Beim Erkennen eines Lade-/Transferfehlers ruft das Systemprogramm den Organisationsbaustein OB32 auf, wenn dieser geladen ist. Der Befehl, der den Transferfehler verursacht hat, wird nicht mehr bearbeitet. Bei Aufruf von OB32 stehen im AKKU 1 zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

LZF -Lade-/Transferfehler (TRAF)

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A11h	-	Zugriff mit U/UN D, O/ON D, S/R D, =D auf ein nicht definiertes Datenwort
1A12h	-	Transferfehler bei TDR auf ein nicht definiertes Datenwort
1A13h	-	Transferfehler bei TDL auf ein nicht definiertes Datenwort
1A14h	-	Transferfehler bei TDW auf ein nicht definiertes Datenwort
1A15h	-	Transferfehler bei TDD auf ein nicht definiertes Datenwort
1A16h	-	Ladefehler bei LDR auf ein nicht definiertes Datenwort
1A17h	-	Ladefehler bei LDL auf ein nicht definiertes Datenwort
1A18h	-	Ladefehler bei LDW auf ein nicht definiertes Datenwort
1A19h	-	Ladefehler bei LDD auf ein nicht definiertes Datenwort
1A1Ah	-	Zugriff mit BDW auf ein nicht definiertes Datenwort

Sonstige Laufzeitfehler (OB31)

Hierzu gehören alle Laufzeitfehler, die nicht einer der vorherigen Laufzeitfehlerarten (Transferfehler oder Aufruf eines nicht geladenen Bausteins) zugeordnet werden können.

Beim Erkennen eines dieser Laufzeitfehler ruft das Systemprogramm den Organisationsbaustein OB31 auf. Der den Fehler verursachende Befehl (bzw. die Sonderfunktion) wird nicht weiter bearbeitet. Wenn der OB31 nicht geladen ist, geht die CPU in den Stoppzustand.

Soll die Programmbearbeitung bei Auftreten eines der nachfolgend aufgeführten Fehler weiterlaufen, genügt die Bausteinende-Anweisung BE im OB31.

Bei Aufruf des OB31 stehen in AKKU 1 und AKKU 2 zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern.

Fehleranzeigen von verschiedenen Operationen, OB254/255 und OB250

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung	
1A21h	-	E DB,EX DX:	Datenbaustein existiert bereits
1A22h	-	E DB, EX DX:	unzulässige Anzahl Datenwörter (<1 oder > 4091)
1A23h	-	E DB, EX DX:	Speicherplatz im RAM reicht nicht aus
1A25h	-	BI:	unzulässiger Parameter im AKKU 1 (<1 oder > 125)
1A29h	-	Klammerstackunter- oder -überlauf nach ‘U(‘; ‘O(‘; ‘)’	
1A2Ah	-	A DB,AX DX:	Bausteinlänge im Datenbausteinkopf ist zu klein (Länge < 5 Wörter)
1A2Bh	-	Funktionsbaustein ist mit falscher PG-Software geladen	
1A2Ch	-	ACR:	Kachelnummer in AKKU-1-L ist unzulässig (>255)
1A31h	-	OB254 bzw. OB255 (Verschieben) oder DB 250:	Ziel-Datenbaustein ist bereits im DB-RAM vorhanden
1A32h	-	OB254 bzw. OB255 (Duplizieren):	Ziel-Datenbaustein ist bereits im DB-RAM vorhanden
1A33h	-	OB254 bzw. OB255 oder OB250:	Speicherplatz im DB-RAM reicht nicht aus

Fehleranzeigen von OB182

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A34h	0001h	Beschreibung des Datenfeldes ist unzulässig
1A34h	0100h	Adressbereichs-Typ ist unzulässig
1A34h	0101h	Datenbaustein-Nr. ist unzulässig
1A34h	0102h	"Nummer des ersten Parameterwortes" ist unzulässig
1A34h	0200h	"Quelldatenbaustein-Typ" ist unzulässig
1A34h	0201h	"Quelldatenbaustein-Nummer" ist unzulässig
1A34h	0202h	"Nummer des ersten zu übertragenden Datenwortes in der Quelle" ist unzulässig
1A34h	0203h	Als Länge des Quelldatenbausteins ist im Bausteinkopf ein Wert < 5 Wörter eingetragen
1A34h	0210h	"Zieldatenbaustein-Typ" ist unzulässig
1A34h	0211h	"Zieldatenbaustein-Nummer" ist unzulässig
1A34h	0212h	"Nummer des ersten zu übertragenden Datenwortes im Ziel" ist unzulässig
1A34h	0213h	Als Länge des Zieldatenbausteins ist im Bausteinkopf ein Wert < 5 Wörter eingetragen
1A34h	0220h	"Anzahl zu übertragender Datenworte" ist unzulässig
1A34h	0221h	Quelldatenbaustein ist zu kurz
1A34h	0222h	Zieldatenbaustein ist zu kurz
1A34h	0223h	Zieldatenbaustein ist im EPROM gespeichert

Fehleranzeigen verschiedener Sonderfunktions-OBs

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung	
1A35h	-	OB250:	Nummer des Übergabebausteins ist unzulässig
1A36h	-	OB250:	unterschiedliche Länge bei DB x und DB x + 1 bzw. DX x und DX x + 1
1A3Ah	-	OB221:	unzulässiger Wert für die neue Zykluszeit (Zykluszeit <1 ms oder > 13000 ms)
1A41h	-	OB240, OB241 oder OB242:	Schieberegister- oder Datenbaustein-Nummer ist unzulässig (Nr. <192 oder > 255)
1A42h	-	OB241:	Schieberegister ist nicht initialisiert
1A43h	-	OB240:	Speicherplatz im DB-RAM reicht nicht aus
1A44h	-	OB240:	Datenwort DW 0 des Datenbausteins hat nicht den Inhalt ‘0’
1A45h	-	OB240:	Schieberegisterlänge in DW 1 ist unzulässig (nicht zwischen 2 und 256)
1A46h	-	OB240:	Zeigerposition ist unzulässig oder Zeigeranzahl ist > 5
1A47h	-	OB120:	Wert in AKKU 1 oder AKKU-2-L-ist unzulässig
1A48h	-	OB122:	Wert in AKKU 1 oder AKKU-2-L-ist unzulässig
1A49h	-	OB110:	Wert in AKKU 1 oder AKKU-2-L-ist unzulässig
1A4Ah	-	OB121:	Wert in AKKU 1 oder AKKU-2-L-ist unzulässig
1A4Bh	-	OB123:	Wert in AKKU 1 ist unzulässig

Fehleranzeigen von OB150

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A4Ch	0001h	Funktionsnummer ist unzulässig (=0 oder >2)
1A4Ch	0100h	Adressbereichs-Typ ist unzulässig
1A4Ch	0101h	Datenbaustein-Nr. ist unzulässig
1A4Ch	0102h	"Nummer des ersten Datenfeldwortes" ist unzulässig
1A4Ch	0103h	als Länge des Datenbausteins ist im Bausteinkopf ein Wert <5 Wörter eingetragen
1A4Ch	0201h	Jahresangabe im Datenfeld ist unzulässig
1A4Ch	0202h	Monatsangabe im Datenfeld ist unzulässig
1A4Ch	0203h	Monatstagangabe im Datenfeld ist unzulässig
1A4Ch	0204h	Wochentagangabe im Datenfeld ist unzulässig
1A4Ch	0205h	Stundenangabe im Datenfeld ist unzulässig
1A4Ch	0206h	Minutenangabe im Datenfeld ist unzulässig
1A4Ch	0207h	Sekundenangabe im Datenfeld ist unzulässig
1A4Ch	0208h	Wert "1/100 Sekunde" im Datenfeld ist ungleich 0
1A4Ch	0209h	Datenfeldwort 3/Bit 0 bis 3 ist ungleich 0
1A4Ch	020Ah	Stunden-Format ist ungleich der Einstellung bei OB151

Fehleranzeigen von OB151, Ob152,OB 153

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
Anzeigen von OB151		
1A4Dh	0001h	Funktionsnummer ist unzulässig (=0 oder >2)
1A4Dh	0100h	Adressbereichs-Typ ist unzulässig
1A4Dh	0101h	Datenbaustein-Nr. ist unzulässig
1A4Dh	0102h	"Nummer des ersten Datenfeldwortes" ist unzulässig
1A4Dh	0103h	als Länge des Datenbausteins ist im Bausteinkopf ein Wert <5 Wörter eingetragen
1A4Dh	0201h	Jahresangabe im Datenfeld ist unzulässig
1A4Dh	0202h	Monatsangabe im Datenfeld ist unzulässig
1A4Dh	0203h	Monatstagangabe im Datenfeld ist unzulässig

Fortsetzung ...

... Fortsetzung

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A4Dh	0204h	Wochentagangabe im Datenfeld ist unzulässig
1A4Dh	0205h	Stundenangabe im Datenfeld ist unzulässig
1A4Dh	0206h	Minutenangabe im Datenfeld ist unzulässig
1A4Dh	0207h	Sekundenangabe im Datenfeld ist unzulässig
1A4Dh	0208h	Wert "1/100 Sekunde" im Datenfeld ist ungleich 0
1A4Dh	0209h	Auftragsart im Datenfeld ist unzulässig (> 7)
1A4Dh	020Ah	Stunden-Format ist ungleich der Einstellung bei OB150
Anzeigen von OB152		
1A4Eh	0001h	Funktionsnummer ist unzulässig (ungleich 0 bis 3 oder ungleich 5 bis 9 oder ungleich 15)
Anzeigen von OB153		
1A4Fh	0001h	Funktionsnummer ist unzulässig (=0 oder <0)
1A4Fh	0002h	Verzögerungszeit unzulässig

Fehleranzeigen von verschied. Systemoperationen

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A50h	-	LRW, TRW: Die errechnete Speicheradresse <BR + Konstante> liegt nicht im Bereich "0 .. EDFFH" [1)]
1A51h	-	LRD, TRD: Die errechnete Speicheradresse <BR + Konstante> liegt nicht im Bereich "0 .. EDFEH" [1)]
1A52h	-	
1A53h	-	
1A54h	-	

Fortsetzung ...

... Fortsetzung

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1A55h	-	LB CB, LW CW, TB CB, TW CW: Die errechnete Kacheladresse <BR + Konstante> liegt nicht im Bereich "F400h .. 77FFh"
1A56h	-	LB CW, LW CD, TB CW, TW CD: Die errechnete Kacheladresse <BR + Konstante> liegt nicht im Bereich "F400h .. F7FFh"
1A57h	-	LB CD, TB CD: Die errechnete Kacheladresse <BR + Konstante> liegt nicht im Bereich "F400h .. F7FFh"
1A58h	-	TNW, TNB: Der Quellblock liegt nicht vollständig in einem dieser Bereiche: 0000 .. 7FFF Anwenderspeicher [1)] 8000 .. DD7F Datenbaustein-RAM DD80 .. E3FF DB 0 E400 .. E7FF S-Merker E800 .. EDFF Systemdaten (BA, BB, BS, BT, Z, T) EE00 .. EFFF Merker, Prozessabbild F000 .. FFFF Peripherie
1A59h	-	TNW; TNB: Der Zielblock liegt nicht vollständig in einem dieser Bereiche: 0000 .. 7FFF Anwenderspeicher [1)] 8000 .. DD7F Datenbaustein-RAM DD80 .. E3FF DB 0 E400 .. E7FF S-Merker E800 .. EDFF Systemdaten (BA, BB, BS, BT, Z, T) EE00 .. EFFF Merker, Prozessabbild F000 .. FFFF Peripherie

[1)] siehe Kapitel "Speicherzugriffe über absolute Adressen"

ADF (Adressierfehler)

Ein Adressierfehler tritt auf, wenn mit einer Operation ein Ein- oder Ausgang im Prozessabbild angesprochen wird, dem zum Zeitpunkt des letzten NEUSTARTs keine Peripheriebaugruppe zugeordnet war (Baugruppe war nicht gesteckt, defekt oder nicht im Datenbaustein DB 1 der CPU angegeben).

OB25

Das Systemprogramm unterbricht nun die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein OB25 auf. Nach der Bearbeitung des im OB25 enthaltenen Programms wird mit dem nächsten Befehl des unterbrochenen Programms fortgefahren. Die Anweisung, die den ADF verursacht hat, wurde zuvor ausgeführt, jedoch mit einem undefinierten Ein- oder Ausgangswert!

Wenn der OB25 nicht programmiert ist, geht die CPU beim Auftreten eines Adressierfehlers in den Stoppzustand, es sei denn, Sie haben für diesen Fall eine Fortsetzung der Programmbearbeitung im Datenbaustein DX 0 festgelegt.

Die Adressierfehlerüberwachung kann durch entsprechende Programmierung des DX 0 auch ganz unterdrückt werden.

Fehleranzeigen

Als Fehleranzeigen werden vom Systemprogramm übergeben:

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1E40h	yyyyyh	Adressierfehler yyyy = ADF-Adresse

QVZ (Quittungsverzug)

Ein Quittungsverzug tritt auf, wenn sich eine Ein- und Ausgabebaugruppe nach einer Adressierung innerhalb einer bestimmten Zeit nicht mit dem RDY-Signal (Ready) zurückmeldet. Die Ursache des Quittungsverzugs kann ein Defekt auf der Peripheriebaugruppe sein oder das Ziehen der Baugruppe aus dem AG während des Betriebs.

Folgende Quittungsverzugsfehler unterbrechen die Anwenderprogrammbearbeitung und rufen einen entsprechenden Organisationsbaustein auf.

Hinweis!

Wenn die aufgerufenen Organisationsbausteine nicht programmiert sind, wird die Bearbeitung des Anwenderprogramms fortgesetzt.

Ist ein Quittungsverzug aufgetreten, liest die CPU "ersatzweise" den Wert "00H" ein und arbeitet, falls der QVZ quittiert wird, mit diesem Wert weiter.

Durch einen Quittungsverzug wird die Laufzeit des Anwenderprogramms verlängert.

STOP bei QVZ

Wenn der Quittungsverzug zum STOP der CPU führen soll, muss im aufgerufenen OB (OB23 bzw. OB24) der Stoppbefehl STP programmiert sein.

Durch entsprechende Programmierung des DX 0 können Sie im Falle eines QVZ auch bei nicht programmierten OB23/24 einen Systemstopp veranlassen.

QVZ bei Direktzugriff über Rückwand-Bus

Quittungsverzug im Anwenderprogramm bei Direktzugriff über den Rückwand-Bus auf CP, IP oder auf eine Peripheriebaugruppe (z.B. mit Lade- und Transferbefehlen L/T P...bzw. Q...).

OB23

Das Systemprogramm ruft den Organisationsbaustein OB23 auf, wenn dieser geladen ist.

Fehleranzeigen

In den AKKUs 1 und 2 stehen dabei zusätzliche Informationen, die den auftretenden Fehler näher erläutern:

QVZ-Anzeigen bei Quittungsverzug

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1E23h	yyyyyh	Quittungsverzug yyyy = QVZ-Adresse

QVZ-Adresse

Die QVZ-Adresse weist auf das Peripheriebyte, welches als erstes einen QVZ erzeugt hat. In der Regel ist dies das Byte mit der niedrigsten Adresse bei Peripheriebefehlen.

Eine Ausnahme hiervon sind QVZ-Adressen, die bei den Befehlen TNB/TNW im Fall eines Quittungsverzuges geliefert werden:

Da diese Befehle dekrementierend arbeiten, zeigt in diesem Fall die QVZ-Adresse auf das Byte mit der höchsten Adresse, das bei einem Blocktransfer den QVZ ausgelöst hat.

QVZ bei PAE/PAA-Aktualisierung Quittungsverzug bei Zugriffen vom Systemprogramm zur Aktualisierung des Prozessabbildes.

OB24 Das Systemprogramm ruft den Organisationsbaustein OB24 auf, wenn dieser geladen ist.

Fehleranzeigen In den AKKUs 1 und 2 stehen zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern.

QVZ-Anzeigen bei Aufruf des OB 24

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung
1E25h	yyyyyh	Quittungsverzug bei der Ausgabe des Prozessabbildes der digitalen Ausgänge yyyy = Adresse des nicht quittierten Ausgangsbytes
1E26h	yyyyyh	Quittungsverzug beim Aktualisieren des Prozessabbildes der digitalen Eingänge yyyy = Adresse des nicht quittierten Eingangsbytes

ZYK (Zykluszeitfehler)

Die Zykluszeit umfasst die gesamte Zeitdauer einer Bearbeitung des zyklischen Programms. Eine Überschreitung der in der CPU eingestellten Zyklusüberwachungszeit kann ausgelöst werden z.B. durch fehlerhafte Programmierung, durch eine Programmschleife in einem Funktionsbaustein, durch Ausfall des Taktgenerators oder durch Systemleistungen wie z.B. Prozessabbildaktualisierung in Verbindung mit langen Programmen.

OB26

Wenn eine Zykluszeitüberschreitung auftritt, unterbricht das Systemprogramm die Bearbeitung des Anwenderprogramms und ruft den Organisationsbaustein OB26 auf, wenn dieser geladen ist. Die Überwachungszeit wird dabei neu gestartet (getriggert). Falls die Überwachungszeit erneut abläuft, bevor der OB26 zu Ende bearbeitet ist, geht die CPU mit Doppelfehler in den Stoppzustand.

Zyklusüberwachungszeit

Die Zyklusüberwachungszeit ist variabel (1 bis 13000 ms) und nachtriggerbar (siehe oben). Unabhängig von der Zykluszeit wird 100 ms nach Ablauf der Zyklusüberwachungszeit BASP aktiv gesetzt, wenn der OB26 zu diesem Zeitpunkt noch nicht zu Ende bearbeitet ist.

Die Zyklusüberwachungszeit können Sie individuell vorgeben durch einen Eintrag im DX 0 oder durch Aufruf des Sonderfunktions-Organisationsbausteins OB221.

Im zyklischen Programm kann die Zyklusüberwachungszeit durch einen Aufruf des Sonderfunktionsbaustein OB222 "nachgetriggert" werden.

STOP bei nicht geladenem OB26

Wenn der OB26 nicht programmiert ist, geht die CPU in den Stoppzustand. Soll dies nicht geschehen, so müssen Sie die Voreinstellung im DX 0 entsprechend ändern.

Fehleranzeigen

Bei Auftreten eines Zykluszeitfehlers werden keine Fehlerkennungen in AKKU 1 oder AKKU 2 übergehen.

WECK-FE (Weckfehler)

Wenn für einen bestimmten Weckalarm-OB eine erneute Anforderung auftritt, bevor seine letzte Anforderung vollständig bearbeitet ist, erkennt das Systemprogramm einen Weckfehler.

OB33

Wenn ein Weckfehler auftritt, ruft das Systemprogramm den Organisationsbaustein OB33 auf, wenn dieser geladen ist, oder die CPU geht in den Stoppzustand. Beachten Sie hierzu auch die Aussagen unter "Weckfehler" im Teil "Betriebszustände und Programmbearbeitungsebenen".

Fehleranzeigen

In den AKKUs 1 und 2 hinterlegt das Systemprogramm zusätzliche Informationen, die den aufgetretenen Fehler näher erläutern:

Fehlerkennung AKKU-1-L	AKKU-2-L	Erläuterung	
1001h	0016h	Weckfehler bei OB10	(10ms)
1001h	0014h	Weckfehler bei OB11	(20 ms)
1001h	0012h	Weckfehler bei OB12	(50 ms)
1001h	0010h	Weckfehler bei OB13	(100 ms)
1001h	000Eh	Weckfehler bei OB14	(200 ms)
1001h	000Ch	Weckfehler bei OB15	(500 ms)
1001h	000Ah	Weckfehler bei OB16	(1 s)
1001h	0008h	Weckfehler bei OB17	(2 s)
1001h	0006h	Weckfehler bei OB18	(5 s)

Hinweis!

Die Kennung im AKKU 2 ist die Ebenenkennung des fehlererzeugenden Weckalarms.

Ist der OB33 nicht programmiert, so geht die CPU in den Stoppzustand. Durch entsprechende Programmierung im DX 0 können Sie bei aufgetretenem Weckfehler und nicht programmiertem OB33 die Programmbearbeitung jedoch weiterlaufen lassen.

Ein erneuter Aufruf der bereits aktivierten Fehlerprogrammbearbeitungsebene "Weckfehler" führt nicht zu einem Doppelfehler (DOPP)!

ABBR (Abbruch)

Auslösung und Reaktion

Wenn im Betriebszustand RUN der Stoppzustand angefordert wird durch

- Betätigen des Betriebsartenschalters an der CPU von RUN auf STOP,
- PG-Online-Funktion AG-STOP,
- Schalter am Koordinator auf STOP,
- so ruft das Systemprogramm den OB28 auf, wenn dieser geladen ist. Nach der Bearbeitung des OB28 geht die CPU in den Stoppzustand.

Hinweis!

Der Übergang in den Stoppzustand erfolgt unabhängig davon, ob und wie der OB28 programmiert ist.

keine Fehlerkennungen

Es werden keine Fehlerkennungen in AKKU 1 oder AKKU 2 übergeben

Kommunikations-fehler (FE-3)

Treten auf der zweiten seriellen Schnittstelle bei Rechnerkopplung RK 512, Datenübertragung mit Prozedur 3964/3964R, Störungen auf, liegt ein Kommunikationsfehler vor. Bei einem Kommunikationsfehler ruft das Systemprogramm den Organisationsbaustein OB35 auf und übergibt in AKKU 1 zusätzliche Informationen, die die aufgetretenen Störungen näher erläutern.

OB35

Haben Sie keinen OB35 programmiert, so erfolgt keine Reaktion des Systemprogramms und die CPU geht nicht in den Stoppzustand. Das ist die Standardeinstellung.

Soll beim Auftreten eines Schnittstellenfehlers die CPU auch bei nicht vorhandenem OB35 in den Stoppzustand gehen, so müssen Sie die Voreinstellung im DX 0 entsprechend ändern.

Fehlerinformation in AKKU 1

Alle 100 ms prüft das Systemprogramm, ob Kommunikationsfehler an der zweiten seriellen Schnittstelle aufgetreten sind. Ist dies der Fall, so hinterlegt das Systemprogramm im AKKU 1 Fehlerinformationen. Ist der OB35 vorhanden, so wird er vom Systemprogramm aufgerufen und die Fehlerinformation in AKKU 1 übergeben.

Es können Fehlernummern zu maximal drei Störungsursachen beim Aufruf des OB35 übergeben werden. Liegen gleichzeitig mehr als drei Störungsursachen vor, so wird dies durch eine spezielle Überlaufkennung angezeigt.

Aufbau der Fehlerinformation in AKKU 1

	31							24	23 18	15 8	7 8
AKKU 1	0	0	0	0	F	U	B	0	Fehlernummer 1	Fehlernummer 2	Fehlernummer 3

F = ‘0’, wenn kein Fehlereintrag im Fehlerbereich
= ‘1’, wenn Fehler in Fehlerbereich eingetragen

U = ‘0’, wenn kein Fehlerüberlauf (maximal drei Einträge)
= ‘1’, wenn Fehlerüberlauf (mehr als drei Einträge)

B = ‘0’, wenn kein BREAK auf der Schnittstelle
= ‘1’, wenn BREAK auf der Schnittstelle

BREAK Bei BREAK auf der Schnittstelle wird der OB35 nur zu Beginn des BREAK-Zustandes aufgerufen.

Fehlernummer 1 bis 3 Hier werden maximal 3 Fehlernummern zu den auf der Schnittstelle erkannten Störungen eingetragen und zwar in der Reihenfolge, in der sie vom System erkannt werden.

Bedeutung der Fehlernummern Die Bedeutung der Fehlernummern sowie weitere Informationen zur Behandlung von Schnittstellenfehlern entnehmen Sie bitte dem Handbuch Ihrer Kommunikationsbaugruppe.

Teil 7 Erweiterter Datenbaustein DX 0

Überblick

In diesem Kapitel wird der Aufbau und die Einträge im erweiterten Datenbaustein DX 0 beschrieben.

Sie erfahren, wofür Sie den Datenbaustein DX 0 einsetzen können und wie er aufgebaut ist. Weiterhin werden Sie informiert welche Bedeutung die verschiedenen DX-0-Parameter haben. Sie lernen an Hand von Beispielen, wie Sie einen DX 0 erstellen bzw. über eine Maske parametrieren können.

Nachfolgend sind beschrieben:

Einstellungen im DX 0

Formaler Aufbau und Parameter

Beispiele

Inhalt

Thema **Seite**

Teil 7 Erweiterter Datenbaustein DX 0 7-1

Anwendung 7-2

Aufbau des DX 0 7-3

Parameter für DX 0 7-5

Parametrierbeispiele 7-9

Anwendung

Als Anwender können Sie bestimmte Leistungen des Systemprogramms Ihren Erfordernissen anpassen, wenn Sie im DX 0 alternativ zu den Standard-Voreinstellungen durch Parametereingabe andere Einstellungen vornehmen.

Die Voreinstellungen des Systemprogramms (in der Parameter-Tabelle mit "V" gekennzeichnet) werden automatisch bei jedem Neustart gesetzt. Danach wird der DX 0 ausgewertet. Haben Sie den Baustein DX 0 nicht geladen, so gelten weiterhin die Voreinstellungen; ansonsten werden die Einstellungen gültig, mit denen Sie den DX 0 parametriert haben.

Sie können die Einstellungen im DX 0 vornehmen, indem Sie die Werte wie bei anderen Datenbausteinen über Anweisungen programmieren oder - mit PG-Systemsoftware S5-DOS von Siemens ab Version 3.0 - die Werte über eine spezielle Maske an Ihrem PG als Parameter eingeben.

Hinweis!

DX 0 wird nur bei Neustart ausgewertet.

Für alle nicht im DX 0 eingetragenen Parameter werden die Voreinstellungen des Systemprogramms übernommen.

Aufbau des DX 0

Einleitung

Der DX 0 setzt sich aus drei Teilen zusammen:

- der Anfangskennung für den DX 0 (DW0, 1 und 2),
- mehreren Blöcken unterschiedlicher Länge (je nach Parameteranzahl),
- der Endekennung EEEEh.

Anfangskennung

ASCII-Zeichen MASKX0 in DW0 bis DW2.

Block

Ein Block im DX 0 besteht aus 1 bis n Datenwörtern. Diese enthalten:

- die Blockkennung,
- die Blocklänge
- und die Blockparameter.

Die Blockkennung gibt an, welche Bedeutung die folgenden Parameter haben. Jeder Block ist einem bestimmten Systemprogrammteil oder einer bestimmten Systemfunktion zugeordnet (z.B. leitet die Blockkennung '04' den Parameterblock für zyklische Programmbearbeitung ein).

Blocklänge

Die Blocklänge gibt an, wieviele Datenwörter die nachfolgenden Parameter belegen.

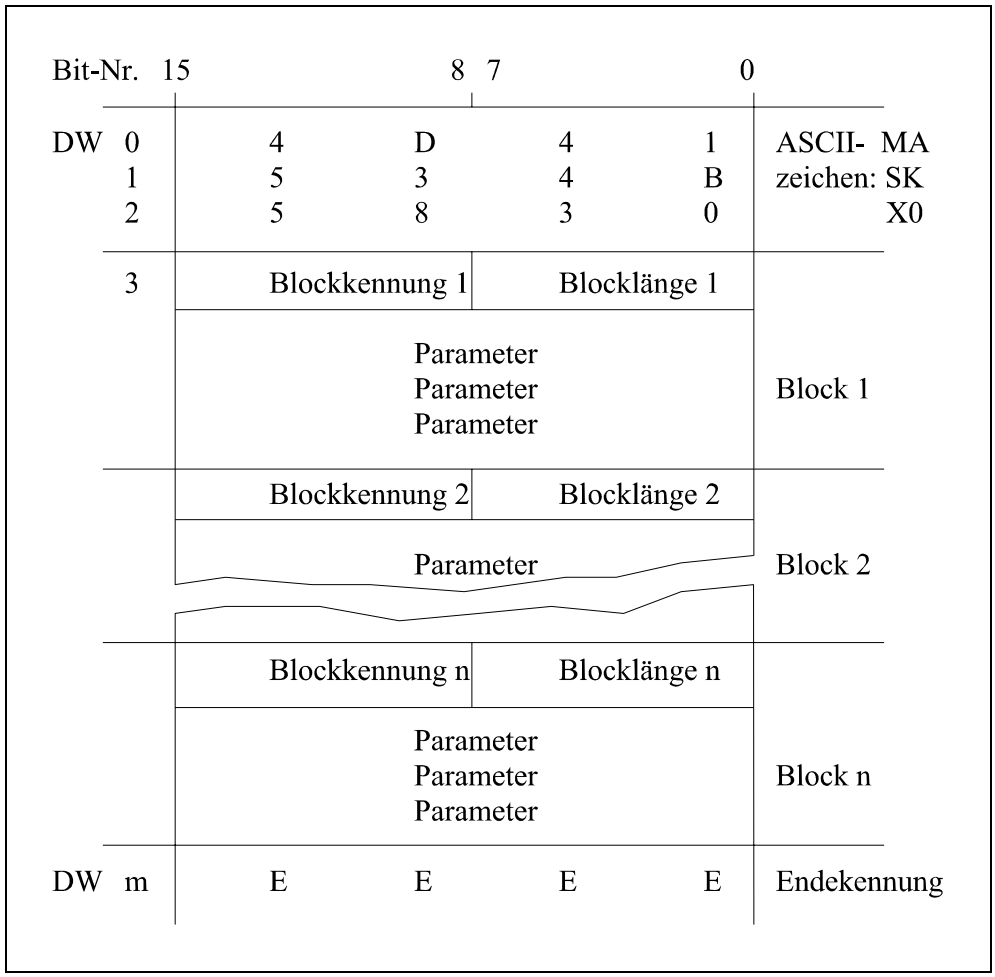
Parameter

Die möglichen Parameter finden Sie im Teil "Parameter des DX0. Die angegebenen Zahlenwerte sind im Hexadezimalformat (KH).

Endekennung

Sie kennzeichnet das Ende des DX 0 durch EEEEh im letzen Datenwort.

Formaler Aufbau des DX 0



Beispiel für die Eingabe des DX 0

Anfangskennung	DW	0:	KH =	4D41	
	DW	1:	KH =	534B	
	DW	2:	KH =	5830	
Blockkennung/-länge	DW	3:	KH =	0201	Block 1
Parameter (belegt 1 DW)	DW	4:	KH =	1001	
Blockkennung/-länge	DW	5:	KH =	0402	
Parameter (belegt 2 DW)	DW	6:	KH =	1000	Block 2
	DW	7:	KH =	0400	
Endekennung	DW	10:	KH =	EEEE	

Beachten Sie bei der Parametrierung des DX 0 folgende Punkte:

- Die Reihenfolge, in der Sie die einzelnen Blöcke eingeben, ist beliebig.
- Nicht benötigte Blöcke brauchen nicht angegeben zu werden.
- Ist ein bestimmter Block mehrfach vorhanden, gilt der Block, den Sie zuletzt eingegeben haben.
- Die Reihenfolge, in der Sie die einzelnen Parameter eingeben, ist beliebig.
- Nicht benötigte Parameter brauchen nicht angegeben zu werden.
- Ist ein bestimmter Parameter mehrfach genannt, gilt die letzte Angabe.

Parameter für DX 0

Bedeutung der Parameter

Blockkennung/ -Länge	Parameter 1. Wort/2. Wort	Bedeutung: (V=Voreinstellung bei nicht geladenem DX 0 bzw. fehlendem Block/Parameter)
ANLAUF und RUN		
02xx [1]	1000 1001	V AUTOMATISCHER WIEDERANLAUF nach NETZ EIN AUTOMATISCHER NEUSTART nach NETZ EIN
	2000 2001	V Synchronisation ANLAUF Keine Synchronisation ANLAUF
	3000 3001	V Adressierfehlerüberwachung Keine Adressierfehlerüberwachung
	4000	V WIEDERANLAUF
	6000 6001	V Gleitpunktarithmetik mit 16-bit-Mantisse (CPU rechnet immer mit 24 bit) Gleitpunktarithmetik mit 24-bit-Mantisse
	BB00 yyyy	Anzahl der zu aktualisierenden Zeitzellen [2] Voreinstellung: yyyy = 256 Zeitzellen, d.h. Zeitzelle 0 bis 255 zulässig: 0...255
	F00X	Identifikation des CPU-Typs 0 = CPU 928 3UB12 1 = CPU 928 3UB21 2 = CPU 928 3UC11 Wenn Eintrag vorhanden ist dann muss dieser mit CPU-Typ übereinstimmen, ansonsten CPU-STOP mit DX0-Fehler!
Zyklische Programmbearbeitung		
04xx [1]	1000 yyyy	Länge der Zyklusüberwachungzeit in Millisekunden; Voreinstellung: yyyy = 150 ms, zulässig: 1 ≤ yyyy ≤ 32C8 (hex) 1 ms bis 13000 ms (dez)
	4000 4001	V Aktualisierung des Prozessabbildes ohne Semaphorschutz Aktualisierung des Prozessabbildes mit Semaphorschutz
Alarmbearbeitung: Weckalarme		
06xx [1]	[3]	Auswahl des Bearbeitungs-Modus [3]
	2000 2001	V Prozessalarm-Signal, pegelgetriggert Prozessalarm-Signal, flankengetriggert
Fehlerbehandlung		
10xx [1]	 1000 1001	Weckfehlerbehandlung: V Systemstopp, wenn Ereignis eintritt und der OB33 nicht geladen ist. Kein Systemstopp, wenn Ereignis eintritt und der OB33 nicht geladen ist.

Fortsetzung ...

... Fortsetzung

Blockkennung/ -Länge	Parameter 1. Wort/2. Wort	Bedeutung: (V=Voreinstellung bei nicht geladenem DX 0 bzw. fehlendem Block/Parameter)
10xx [1]		
		Zyklusfehlerbehandlung:
	1400	V Systemstopp, wenn Ereignis eintritt und der OB26 nicht geladen ist.
	1401	Kein Systemstopp, wenn Ereignis eintritt und der OB26 nicht geladen ist.
		Befehlscodefehlerbehandlung:
	1800	V Systemstopp, wenn Ereignis eintritt und der OB27/29/30 nicht geladen ist.
	1801	Kein Systemstopp, wenn Ereignis eintritt und der OB27/29/30 nicht geladen ist.
		Laufzeitfehlerbehandlung:
	1A00	V Systemstopp, wenn Ereignis eintritt und der OB19/31/32 nicht geladen ist.
	1A01	Kein Systemstopp, wenn Ereignis eintritt und der OB19/31/32 nicht geladen ist.
		Adressierfehlerbehandlung:
	1C00	V Systemstopp, wenn Ereignis eintritt und der OB25 nicht geladen ist.
	1C01	Kein Systemstopp, wenn Ereignis eintritt und der OB25 nicht geladen ist.
		Quittungsverzugsfehlerbehandlung:
	1E00	Systemstopp, wenn Ereignis eintritt und der OB23/24 nicht geladen ist.
	1E01	V Kein Systemstopp, wenn Ereignis eintritt und der OB23/24 nicht geladen ist.
		Schnittstellenfehlerbehandlung:
	2000	Systemstopp, wenn Ereignis eintritt und der OB35 nicht geladen ist.
	2001	V Kein Systemstopp, wenn Ereignis eintritt und der OB35 nicht geladen ist.
EEEE		Endekennung

[1] xx = Blocklänge (Anzahl der von den Parameter belegten Wörter)

[2] Zur Aktualisierung der Zeitzellen lesen Sie bitte die Erklärung auf der nächsten Seite

[3] Parameter und Bedeutung entnehmen Sie bitte der auf Seite 7-8 aufgeführten Tabelle

Parametereingabe

Die aktuelle PG-Software (STEP® 5/ST Vers. 6 bzw. STEP® 5/MT Vers. 2 von Siemens) zum Generieren des DX 0 mit Hilfe einer Maske setzt die Parameter für die Schnittstellenfehlerbehandlung (Blockkennung 10xx, Parameter 2000 bzw. 2001) nicht ab.

Sie können diese Parameter z.B. mit der PG-Funktion "Ausgabe Baustein" eintragen (vergessen Sie dabei nicht, die Blocklänge zu ändern!). Ein so geänderter DX 0 ist mit der aktuellen PG-Software mit Hilfe der Funktion "Ausgabe Maske" nicht mehr editierbar.

Aktualisierung der Zeitzellen

- Standardmäßig werden die Zeitzellen T 0 bis T 255 aktualisiert.
- Wenn Sie im DX 0 die Anzahl '0' eintragen, werden keine Zeitzellen aktualisiert, auch dann nicht, wenn Sie im Programm enthalten sind. Es gibt dann auch keine Fehlermeldung.

Parameter

für die Alarmbearbeitung

Aus der folgenden Tabelle können Sie für Ihre Alarmbearbeitung den passenden Parameter heraussuchen und den DX 0 damit programmieren.

Je nachdem, welchen Parameter Sie wählen, sind entsprechend den Symbolen bestimmte (oder alle) Alarme an Bausteingrenzen, andere (oder alle) an Befehlsgrenzen wirksam.

Parameter/ (alt) [1]	Zeitalarm	Weckalarme									Regleralarm	Verz.-Alarm	Proz.-Alarm
		5s	2s	1s	500ms	200ms	100ms	50ms	20ms	10ms			
122C V (100C)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1224 (100A)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	■
1220	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	■	■
121C (1008)	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	■	■	■
1216	☐	☐	☐	☐	☐	☐	☐	☐	☐	■	■	■	■
1214	☐	☐	☐	☐	☐	☐	☐	☐	■	■	■	■	■
1212	☐	☐	☐	☐	☐	☐	☐	■	■	■	■	■	■
1210	☐	☐	☐	☐	☐	☐	■	■	■	■	■	■	■
120E	☐	☐	☐	☐	☐	■	■	■	■	■	■	■	■
120C	☐	☐	☐	☐	■	■	■	■	■	■	■	■	■
120A	☐	☐	☐	■	■	■	■	■	■	■	■	■	■
1208	☐	☐	■	■	■	■	■	■	■	■	■	■	■
1206	☐	■	■	■	■	■	■	■	■	■	■	■	■
1204 (1006)	■	■	■	■	■	■	■	■	■	■	■	■	■

V = Voreinstellung

☐ Unterbrechung an Bausteingrenzen

■ Unterbrechung an Befehlsgrenzen

[1] Die PG-Software zum Generieren des DX 0 setzt die "alten" Parameter ein. Soll ein DX 0, der über STEP® 5 von Siemens mit neuen Parametern erstellt wurde, am PG angezeigt werden, so führt dies zu einer Fehlermeldung.

Hinweis!

Falls die Alarmbearbeitung an Befehlsgrenzen eingeschaltet ist, werden bei Unterbrechungen u.U. auch die Operationen 'TNB' bzw. 'TNW' unterbrochen. Dies gilt ebenso für einige wenige Sonderfunktions-Organisationsbausteine, Standardfunktionsbausteine und Regler-Funktionsbausteine.

Parametrierbeispiele

Programmierung

Beispiel

Mit der folgenden Parametrierung des DX 0 wird

- die Adressierfehler-Überwachung abgeschaltet,
- die Zeitzellenaktualisierung abgeschaltet,
- die Zykluszeit auf 4 s eingestellt.

DX 0	Anfangskennung "MASKX0"	DW0: KH = 4D41
		DW1: KH = 534B
		DW2: KH = 5830
	1. Blockkennung/-länge	DW3: KH = 0203
	Parameter	DW4: KH = 3001
	Parameter [1)]	DW5: KH = BB00
		DW6: KH = 0000
	2. Blockkennung/-länge	DW7: KH = 0402
	Parameter [1)]	DW8: KH = 1000
		DW9: KF = +4000
	Endekennung	DW10: KH = EEEE

[1)] Parameter, die zwei Datenwörter belegen, müssen bei der Angabe der Blocklänge mit '2' berücksichtigt werden!

Diese Parametrierung des DX 0 hat folgende Auswirkungen auf die Programmbearbeitung:

- Derjenige Teil des Prozessabbilds, dem keine Peripheriebaugruppen zugeordnet sind, kann als zusätzlicher "Merkerbereich" benutzt werden.
- Die Laufzeit des Systemprogramms verkürzt sich, da keine Zeitzellen aktualisiert werden.
- Ein Zyklusfehler wird erst dann erkannt, wenn die Laufzeit des Anwenderprogramms und des Systemprogramms zusammen 4 s übersteigt.

Parametrierung über PG-Maske

Für die Parametrierung des DX 0 stehen Ihnen bei der PG-Systemsoftware S5-DOS von Siemens (ab Version 3) Masken zur Verfügung. Die PG-Software generiert entsprechend den Parametervorbesetzungen und den von Ihnen eingegebenen Parametern automatisch den Datenbaustein DX 0. Für die Parametrierung werden zwei Masken benötigt.

Die grundsätzlichen Bedienungen zur Anwahl und zum Ausfüllen von PG-Masken entnehmen Sie bitte Ihrer PG-Beschreibung.

Hantierung beim Ausfüllen der DX 0-Masken

Die PG-Maske zum Ausfüllen des DX 0 besteht aus zwei Teilmasken:

1. Teilmaske DX 0

```
DX0-PARAMETRIERUNG (AG135U: CPU928, R-PROZESSOR)          DX0

ANLAUFART NACH NETZ EIN:                  1     (1 = WIEDERANLAUF
                                                 2 = NEUSTART)

MEHRPROZESSORANLAUF SYNCHRONISIEREN       JA

BLOCKUEBERTRAGUNG DER KOPPELMERKER        NEIN

ADRESSIERFEHLERUEBERWACHUNG               JA

ZYKLUSZEITUEBERWACHUNG (X 10MS):          15    (R-PROZ: 1-400
                                                 CPU 928: 1-600)

ANZAHL DER ZEITZELLEN                     256   (R-PROZ: 0-128
                                                 CPU 928: 0-256

GENAUIGKEIT DER GLEITPUNKTARITHMETIK      16-BIT-MANTISSE
#24-BIT-MANTISSE NUR BEI CPU 928#

 F1  |  F2  |  F3  |  F4  |  F5  |  F6  |  F7  |  F8
      WAEHLEN                    WEITER
```

2. Teilmaske DX 0

```
DX0-PARAMETRIERUNG (AG135U: CPU928, R-PROZESSOR)          DX0

SYSTEMSTOP BEI EREIGNIS UND NICHT VORHANDENEM FEHLER-OB:

ADRESSIERFEHLER   (OB25)          JA      ZYKLUSFEHLER (OB26)  JA

QUITTUNGSFEHLER   (OB23, 24)      NEIN    WECKFEHLER   (OB33)  JA

BEFEHLSCODEFEHLER (OB27, 29, 30) JA      REGLERFEHLER (OB34)  JA

LAUFZEITFEHLER    (OB19, 31, 32) JA

PROZESSALARMBEARBEITUNG           PEGEL     - GETRIGGERT

UNBERECHENBARKEIT DES ANWENDERPROGRAMMS DURCH ALARME:  MODE   1
 1:ALLE ALARME AN BAUSTEINGRENZEN
 2:ALLE ALARME AN BEFEHLSGRENZEN
 3:NUR PROZESSALARME AN BEFEHLSGRENZEN
 4:NUR PROZESS- UND REGLERALARME AN BEFEHLSGRENZEN
 X:(X=10,...17) WECKALARM VON OB10-OBX UND REGLER-/PROZESS-
   ALARM AN BEFEHLSGRENZEN #NUR MOEGLICH BEI CPU 928#

F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8
        WAEHLEN                WEITER
```

Das nachfolgende Ablaufdiagramm erläutert Ihnen, wie Sie die Teilmasken und Maskenfelder ausfüllen, die Parameter abspeichern und den erzeugten Datenbaustein DX 0 laden können.

Ablaufdiagramm zum Ausfüllen der DX 0-Parameter

<table>
<tr><td colspan="3">Sind in der 1.Teilmaske Parameter zu ändern?</td></tr>
<tr><td>NEIN</td><td colspan="2">JA</td></tr>
<tr><td rowspan="4"></td><td colspan="2">Wiederholen Sie folgenden Vorgang so oft, bis Sie alle notwendigen Änderungen in der Teilmaske durchgeführt haben:</td></tr>
<tr><td></td><td>-Eingabefeld anwählen:
Schreibmarke vor das Parameterfeld positionieren. Das Anzeigefeld F3 am unteren Bildschirmrand zeigt an, ob Sie zwischen Alternativen auswählen (Anzeige "WAEHLEN") oder den Parameterwert verändern können (Anzeige "EINGEBEN").</td></tr>
<tr><td></td><td>-WAEHLEN:
Betätigen Sie so oft die Taste F3, bis die gewünschte Alternative angezeigt wird.</td></tr>
<tr><td></td><td>-"EINGEBEN":
Betätigen Sie einmal die Taste F3: die Schreibmarke springt an den Feldanfang. Sie können nun das Feld mit einem erlaubten Zahlenwert überschreiben.</td></tr>
<tr><td colspan="3">Sind in der 2.Teilmaske Parameter zu ändern?</td></tr>
<tr><td>NEIN</td><td colspan="2">JA</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Betätigen Sie die Taste F6 ("WEITER"); es wird Ihnen die 2.Teilmaske angezeigt.</td></tr>
<tr><td colspan="2">Verfahren Sie beim Ändern der Parameter so, wie oben für die 1.Teilmaske geschildert.</td></tr>
<tr><td colspan="3">Betätigen Sie nun die Übernahmetaste: Die PG-Software übernimmt alle Parametereinstellungen aus den beiden Teilmasken und generiert daraus den Datenbaustein DX 0.</td></tr>
<tr><td colspan="3">Der DX 0 ist im PG abgespeichert. Sie können ihn mit dem PG in die CPU laden oder in einem EPROM-Modul abspeichern.</td></tr>
</table>

Beispiel zum Ausfüllen der DX 0-Maske

Sie wollen den DX 0 für folgendes Verhalten des Systemprogramms abweichend von den Vorbesetzungen parametrieren:

- Die Zyklusüberwachungszeit soll 100 ms betragen.
- Arithmetische Operationen sollen mit 24-Bit-Gleitpunktmantisse durchgeführt werden.
- Bei Zyklusfehlern soll die CPU nicht in STOP gehen, wenn der OB26 nicht geladen ist.
- Das Anwenderprogramm soll von allen Alarmen an Befehlsgrenzen unterbrochen werden.

Füllen Sie für diese Wünsche die Maske folgendermaßen aus:

Erste DX 0-Maske:

- Wählen Sie bei Parameter MEHRPROZESSORANLAUF SYNCHRONISIEREN mit der Funktionstaste F3 NEIN an.
- Betätigen Sie beim Parameter ZYKLUSZEITUEBERWACHUNG zunächst die Funktionstaste F3 und geben Sie die Ziffer 10 (= 100 ms) ein.
- Wählen Sie beim Parameter GENAUIGKEIT DER GLEITPUNKTARITHMETIK mit der Funktionstaste F3 die "24-Bit-Mantisse" an.
- Betätigen Sie die Funktionstaste F6 (WEITER). Es wird danach die zweite DX 0-Maske vorgelegt.

Zweite DX 0-Maske:

- Wählen Sie bei Parameter ZYKLUSFEHLER mit der Funktionstaste F3 NEIN an.
- Geben Sie in das Feld MODE bei Parameter UNTERBRECHBARKEIT DES ANWENDERPROGRAMMS DURCH ALARME die Ziffer '2' (= alle Alarme an Befehlsgrenzen) ein.
- Bestätigen Sie mit der Übernahmetaste die gesamten Eingaben. Der Datenbaustein DX 0 wird nun von der PG-Software generiert.

Übertragen Sie anschließend den DX 0 in den Speicher der CPU oder in ein EPROM-Speichermodul.

Teil 8 Speicherbelegung und -organisation

Überblick

In diesem Kapitel wird die Stuktur des Speichers und die Organisation des Anwenderspeichers der CPU 928 beschrieben.

Sie finden hier wichtige und für den Anwender zugängliche Informationen, die in einigen Systemdatenwörtern hinterlegt sind.

Nachfolgend sind beschrieben:

- Adressraumaufteilung
- Bausteinköpfe
- Bausteinadresslisten im DB 0
- BA/BB-Bereich
- BS/BT-Bereich
- Bitbelegung der Systemdatenwörter

Inhalt

Thema **Seite**

Teil 8 Speicherbelegung und -organisation 8-1

Struktur des Speichers CPU 928 8-2

Adressbereiche für die Peripherie und deren Programmierung 8-4

Organisation des Anwenderspeichers CPU 928 8-5

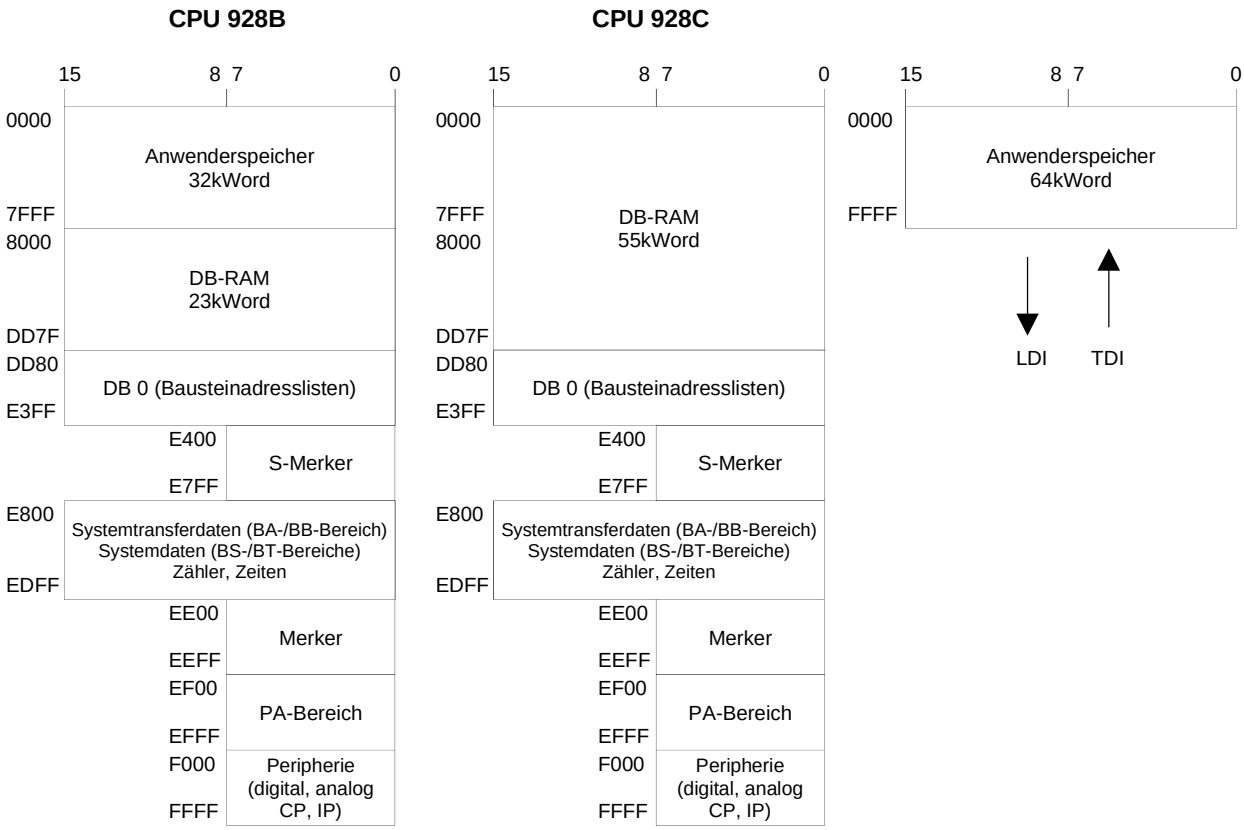
Struktur des Speichers CPU 928

Unterschiede CPU 928B und CPU 928C

Bezüglich der Speicherstruktur unterscheidet sich die CPU 928B von der CPU 928C in der Größe des DB-Speichers. Im Gegensatz zur CPU 928B ist der Anwenderspeicher bei der CPU 928C in eine zusätzliche Speicherbank ausgelagert. Dadurch vergrößert sich das DB-RAM bei der CPU 928C und beginnt schon bei 0000. Der Anwenderspeicher steht nun nicht mehr im direkten Zugriff.

Für den Zugriff auf der Anwenderspeicher in der CPU 928C sind die zwei Befehle LDI und TDI erforderlich, die den LIR- und TIR-Befehl in diesem Bereich ersetzen.

Das nachfolgende Bild soll nochmals die Unterschiede verdeutlichen:



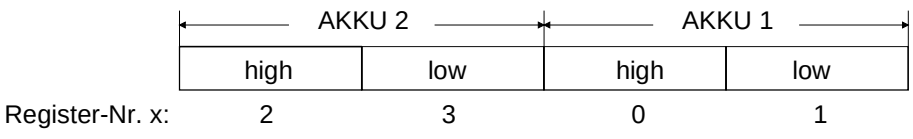
LDI x (nur CPU 928C)

Das Register x mit dem Inhalt eines Anwenderspeicher-Worts laden, dessen Adresse sich in AKKU 1 befindet. Unterstützt werden x = 0 ... 3.

TDI x (nur CPU 928C)

Den Registerinhalt von x in das Anwenderspeicherwort, dessen Adresse sich in AKKU 1 befindet, transferieren. Unterstützt werden x = 0 ... 3.

Register für LDI und TDI



Hinweis!

Der Zugriff auf eine Speicherzelle innerhalb eines Operandenbereichs wie z.B. Merker sollten nie direkt über die absolute Adresse dieser Speicherzelle erfolgen. Verwenden Sie indirekte Zugriffe relativ zur Basisadresse. Die Basisadresse aller Operandenbereiche finden Sie im BS-Bereich.

Im Speicher der CPU 928 finden Sie folgende Bereiche:

Adressraum	Speicherbereich		Länge	Breite
CPU 928B				
0000...7FFF	Anwenderspeicher für:	OB, FB, FX, PB, SB, DB, DX	32 kWord	16 Bit
8000...DD7F	DB-RAM für:	Datenbaustein, Schieberegister	23 kWord	16 Bit

CPU 928C	*Der Anwenderspeicher ist in Speicherbank ausgelagert. Das DB-RAM beginnt schon bei 0000. Für direkten Zugriff auf den Anwenderspeicher sind die mitgelieferten Befehle LDI und TDI erforderlich)*			
0000...FFFF	Anwenderspeicher für:	OB, FB, FX, PB, SB, DB, DX	32 kWord	16 Bit
0000...DD7F	DB-RAM für:	Datenbaustein, Schieberegister	55 kWord	16 Bit

CPU 928B und CPU 928C				
DD80...E3FF	DB 0	Bausteinadresslisten	1664 Byte	16 Bit
E400...E7FF	Merker:	S	1024 Byte	8 Bit
	Bereich Anschaltung:			
E800...E8FF		BA Bereich Anschaltung	256 Worte	16 Bit
E900...E9FF		BB Erweiterter Bereich Ansch.	256 Worte	16 Bit
	Bereich System:			
EA00...EAFF		BS Bereich Systemdaten	256 Worte	16 Bit
EB00...EBFF		BT Erweiterter Ber. Systemdat.	256 Worte	16 Bit
EC00...ECFF	Zähler:	Z	256 Worte	16 Bit
ED00...EDFF	Zeiten:	T	256 Worte	16 Bit
EE00...EEFF	Merker:	M	256 Byte	8 Bit
EF00...EFFF	Prozessabbild der E/A	PAE, PAA	je 128 Byte	8 Bit
F000...FFFF	Peripheriebereich unterteilt in:			
	P-Peripherie		256 Byte	8 Bit
F000...F07F		Digitale Peripherie (mit PA)	128 Byte	8Bit
F080...F0FF		Digitale oder analoge Peripherie (ohne PA)	128 Byte	8 Bit
F100...F1FF	Q-Peripherie		256 Byte	8 Bit
F200...F2FF			256 Byte	8 Bit
F300...F3FF			256 Byte	8 Bit
F400...FBFF	Kacheln	CP, IP,	2048 Byte	8 Bit
FC00...FCFF	IM 3		256 Byte	8 Bit
FD00...FDFF	IM 4		256 Byte	8 Bit
FE00...FEFE	Dezentrale Peripherie	erweitertes Adressvolumen	255 Byte	8 Bit
FEFF	belegt bei Einsatz von Kachelbaugruppen		1 Byte	8 Bit
FF00...FFFF	reserviert		256 Byte	8 Bit

Adressbereiche für die Peripherie und deren Programmierung

Bereich (absolute Adresse)		wird angesprochen mit	Parameter
EF00...EF7F	PAE (Prozessabbild Eingänge)	L EB / T EB	0...127
		L EW / T EW	0...126
		L ED / T ED	0...124
		U E / UN E	0.0...127.7
		O E / ON E	0.0...127.7
		S E / R E / =E	0.0...127.7
EF80...EFFF	PAA (Prozessabbild Ausgänge)	L AB / T AB	0...127
		L AW / T AW	0...126
		L AD / T AD	0...124
		U A / UN A	0.0...127.7
		O A / ON A	0.0...127.7
		S A / R A / =A	0.0...127.7
P-Peripherie mit Prozessabbild Während der Operationsbearbeitung wird nur das Prozessabbild verändert. Erst am Ende des Zyklus wird der Zustand des Prozessabbilds der Ausgänge an die Peripherie ausgegeben.			
F000...F07F	Digitale Peripherie Eingänge/Ausgänge	L PY / T PY	0...127
		L PW / T PW	0...126
F080...F0FF	Digitale oder analoge Peripherie Eingänge/ Ausgänge	L PY / T PY	128...255
		L PW / T PW	128...254
P-Peripherie Die Ein- und Ausgänge werden byte- oder wortweise direkt angesprochen.			
F100...F1FF	Erweiterte Peripherie Eingänge/Ausgänge	L QB / T QB	0...255
		L QW / T QW	0...254
Q-Peripherie Die Ein- und Ausgänge werden byte oder wortweise direkt angesprochen.			

Mit Operationen können Sie entweder direkt oder über das Prozessabbild auf die Peripherie zugreifen.

Hinweis!

Über Anschaltungen wie IM308 können Sie mit Ihrem Programm auf dezentrale Adressbereiche (gleichwertig zum Q-Bereich) zugreifen. Ein Zugriff ist aber nur über absolute Adressierung möglich.

Organisation des Anwenderspeichers CPU 928

Der Anwenderspeicher umfasst, den Speicherbereich bei der CPU 928B 0000h...7FFFh (bei 928C 0000h...FFFFh). Beim Laden werden die einzelnen Bausteine in beliebiger Reihenfolge im Speicher abgelegt.

Die Position des Anwenderspeichers ist von der eingesetzten CPU abhängig. Bei der CPU 928C kommt eine zusätzliche Speicherbank zum Einsatz, so dass das DB-RAM vergrößert wird und schon bei 0000 beginnt.

Hinweis!

Bitte beachten Sie, dass über LIR und TIR kein direkter Zugriff auf den Anwenderspeicher möglich ist. Verwenden Sie hierzu die mitgelieferten Befehle LDI und TDI.

"alternatives Laden" der Datenbausteine

Abhängig von den Einstellungen im Systemdatenwort BS144 erfolgt das Laden der Datenbausteine DB/DX:

Die Datenbausteine werden zuerst in den Anwenderspeicher geladen. Wenn dieser gefüllt ist, werden die Datenbausteine im internen DB-RAM abgelegt. Durch Setzen des Bit 0 in BS144 können Sie die Reihenfolge umkehren: "alternatives Laden".

CPU 928C

Diese Funktion ist bei der CPU 928C nicht erforderlich, da Datenbausteine nur im DB-RAM abgelegt werden.

Speicherauskunft

Mit der PG-Online-Funktion SPAUS (Speicherausbau) erhalten Sie die Hex-Adresse der Speicherzelle die den Bausteinende-Befehl des letzten im Speichermodul vorhandenen Bausteins und die Größe des RAM-Moduls enthält.

CPU 928C

SPAUS sollte beim Einsatz der CPU 928C aufgrund der getrennten Speicherbereiche nicht verwendet werden. Verwenden Sie statt dessen BS64 und BS65, die in diesem Kapitel noch näher beschrieben sind.

Verwaltung der Bausteine

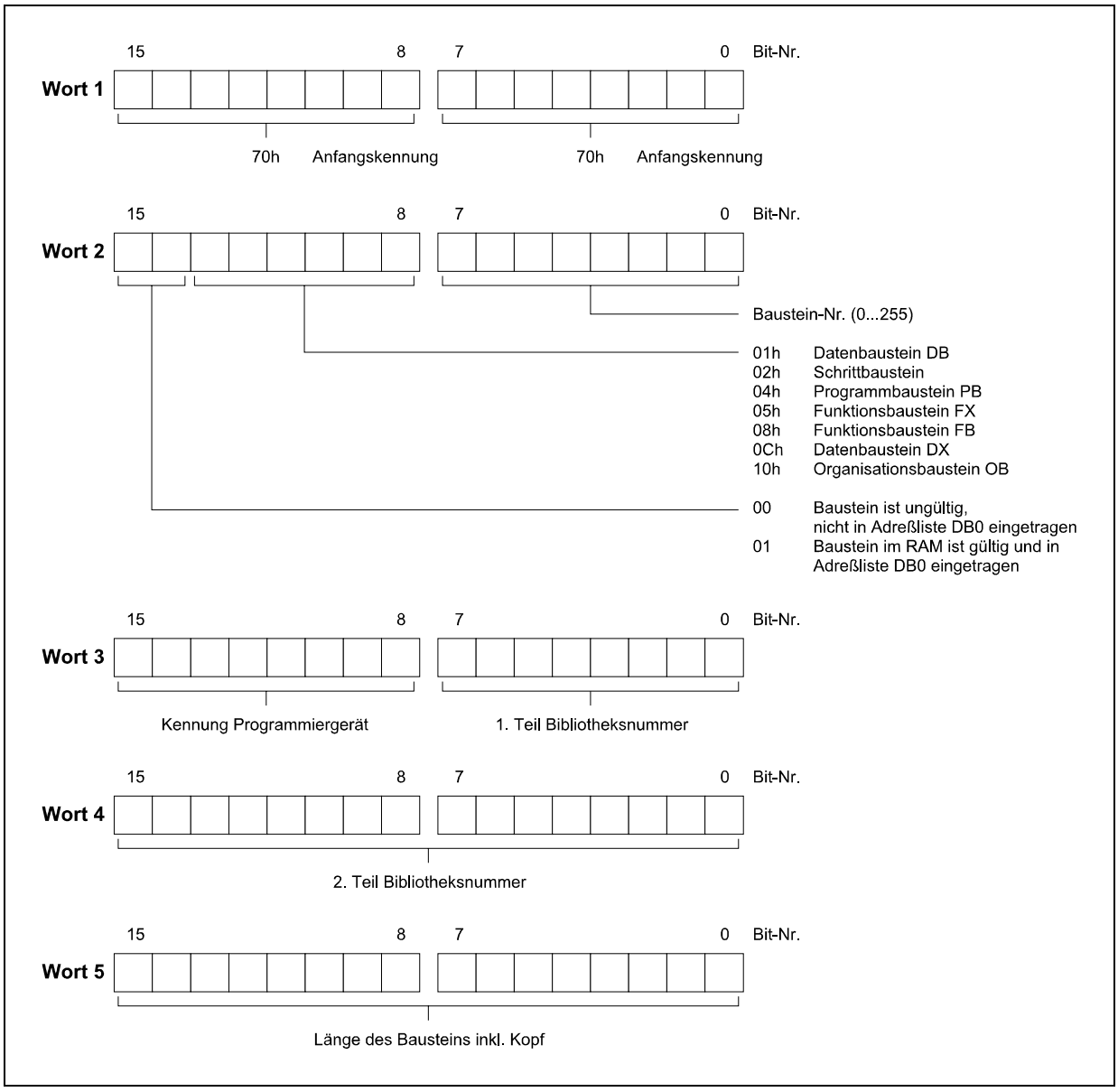
Beim *Korrigieren* von Bausteinen wird der alte Baustein für ungültig erklärt und ein neuer Baustein in der Adressliste und im Speicher eingetragen. Beim *Löschen* werden die gelöschten Bausteine für ungültig erklärt.

Speicher komprimieren

Mit der Online-Funktion SPEICHER KOMPRIMIEREN schaffen Sie Speicherplatz für neue Bausteine. Ungültige Bausteine werden entfernt und gültige Bausteine zusammengeschoben.

Bausteinköpfe im Anwender-speicher

Jeder Baustein im Speicher beginnt mit einem 5 Worte langen Bausteinkopf.



Bausteinadress-listen im DB 0

Der Datenbaustein DB 0 enthält die Anfangsadressen aller Anwender-bausteine, die sich im Anwenderspeicher oder im DB-RAM der CPU befinden. Diese Adressliste wird nach NETZ EIN vom Systemprogramm bei NEUSTART aufgebaut, bei WIEDERANLAUF überprüft und bei jeder Eingabe oder Änderung von Bausteinen mit einem PG automatisch aktualisiert.

Für jeden Bausteintyp gibt es im DB 0 eine eigens reservierte, 256 Worte lange Adressliste, d.h. für jeden Baustein ist 1 Wort reserviert. Nicht geladene und gelöschte Bausteine haben die Anfangsadresse 0.

Die Anfangsadressen der einzelnen Bausteinadresslisten finden Sie außerdem in den Systemdaten:

BS32: DX, BS33: FX, BS34: DB, BS35: SB, BS36: PB, BS37: FB, BS38: OB (nur 48 Worte).

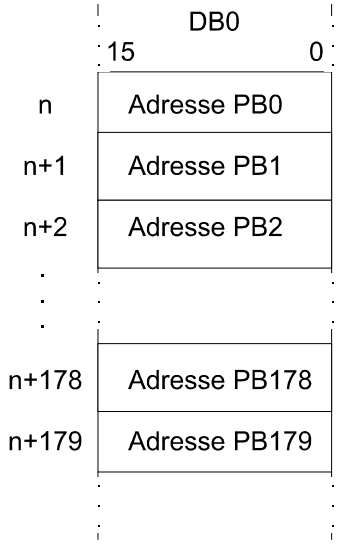
Bei Code-Bausteinen zeigen die Bausteinanfangsadressen jeweils auf die 1. Anweisung. Bei FBs auf den SPA-Befehl vor dem Namen und der Parameterliste. Die Anfangsadressen der Datenbausteine zeigen jeweils auf das Datenwort DW0.

Hinweis!

Die Bausteinanfangsadressen in den Adresslisten zeigen immer auf das erste Wort nach dem Bausteinkopf.

Ablage der Bausteinadressen im DB 0

Folgende Abbildung soll die Ablage der Bausteinadressen im DB 0 verdeutlichen. Hierbei steht n für die Anfangsadresse der PB-Adressliste.



Ist als Adresse der Wert 0 eingetragen, ist der Baustein nicht geladen.

Beispiele für die Ermittlung der Bausteinadresse

Anfangsadresse des FB 40

Lösung a)

```
:L      BS 37  Basisadresse FB-Adressliste
:L      KB 40  +FB-Nummer
:+F            =Adresse der Speicherzelle, die
               die Anfangsadresse des FB 40 enthält
:LIR    1      Anfangsadresse des FB 40 in den
               AKKU 1 laden
               (Baustein ist nicht vorhanden
               wenn Anfangsadresse=0)
```

Lösung b)

```
:L      BS 37  Basisadresse FB-Adressliste
:MAB           BR-Register mit Basisadresse laden
:LRW    +40    Inhalt von Speicherzelle
               (Basisadresse+40) in den AKKU 1 laden
```

Anfangsadresse und Länge des Datenbausteins DB 50 ermitteln

Lösung a) über indirekten Speicherzugriff

```
      :L      BS 34  Basisadresse der DB-Adressliste laden
      :L      KB 50  Die Adresse für den Eintrag für den
                     DB 50 errechnen
      :+F            und die Anfangsadresse
      :LIR    1      in den AKKU 1 laden
      :L      KB 0   Bei nicht vorhandenem Baustein zur
      :!=F           Marke NIVO springen
      :SPB    =NIVO
      :ENT           Anfangsadresse des DB 50 in AKKU 3
                     und
      :TAK           AKKU 1 laden
      :L      KF -1  Anfangsadresse um 1 vermindern und
      :+F            Bausteinlänge in den AKKU 1 laden
      :LIR    1
      : .
NIVO  :.........     Reaktion, wenn Baustein nicht vor-
                     handen
```

Lösung b) über OB 181, Datenbausteine DB/DX testen

```
      :L      KY1,50 Datenbaustein DB 50
      :SPA    OB 181 Datenbausteine DB/DX testen
      :SPB    =NIVO  Sprung, wenn nicht vorhanden
      :SPM    =PROM  Sprung, wenn im EPROM-Modul
      :SPZ    =ANWE  Sprung, wenn im RAM-Modul
      :SPP    =DBRA  Sprung, wenn im DB-RAM
      :SPA    =Fehl  Sprung auf Fehlerbearbeitung
NIVO  :              Baustein nicht vorhanden
      :BEA
PROM  :              DB liegt in Anwenderspeicher (EPROM)
      :BEA
ANWE  :              DB liegt im Anwenderspeicher (RAM)
      :BEA
DBRA  :              DB liegt im DB-RAM
      :BEA
FEHL  :              Fehlerbearbeitung
      :BEA
      :BE
```

BA/BB-Bereich

BA-Bereich

Der BA-Bereich ist ein 256 Worte langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E800h bis E8FFh.

BB-Bereich

Der BB-Bereich ist ein 256 Worte langer Bereich im internen System-RAM der CPU. Er belegt die Adressen E900h bis E9FFh.

Sie können den BA- und den BB-Bereich für eigene Zwecke nutzen.

Hinweis!

Bitte beachten Sie dass diese Bereiche beim Urlöschen mit 0 vorbesetzt werden!

BS/BT-Bereich

In diesen Bereichen finden Sie Informationen für den Systemprogrammierer und systeminterne Daten.

BS-Bereich

Der BS-Bereich ist ein 256 Worte langer Bereich im internen System-RAM der CPU. Er belegt die Adressen EA00h bis EAFFh.

Achtung!

Sie dürfen ausschließlich die Systemdatenwörter BS1, BS60...BS63, BS133, BS140 und BS144 beschreiben!

Alle übrigen Systemdaten dürfen nur gelesen werden. Die Beschreibung dieser Systemdaten kann zu schweren Systemstörungen in Ihrem Automatisierungssystem führen und Mensch und Maschine in Gefahr bringen!

BS60...BS63	stehen für Ihre eigenen Zwecke zur Verfügung
BS1, BS133	haben eine festgelegte Bedeutung und beeinflussen die Programmbearbeitung. Sie dürfen nur mit gültiger Kennung beschrieben werden.

BT-Bereich

Der BT-Bereich ist ein 256 Worte langer Bereich im internen System-RAM der CPU. Er belegt die Adressen EB00h bis EBFFh.

Den gesamten BT-Bereich (BT0 bis BT255) können Sie für Ihre eigenen Zwecke nutzen.

Der BS-/BT-Bereich wird nur beim Urlöschen gelöscht.

Informationen zu Systemdaten erhalten Sie auch über die Online-Funktion SYSTEM-PARAMETER.

Die nachfolgende Tabelle zeigt die Belegung des BS-Bereichs:

Belegung des BS-Bereichs

BS	Bezeichnung		Adr
0	Unterbrechungsanzeigewort (STOER.URS)		EA00
1	Unterbrechungsanzeigenloeschwort (UALW)		EA01
2	Unterbrechungsanzeigensammelwort (UAMK)		EA02
3 Anlauf-Fehlererkennung-Anzeige 4			EA03 EA04
5	Stopp-Kennung	Anlauf-Kennung	EA05
6	Zyklus-Kennung	Modul-Kennung	EA06
7	Urlösch-Kennung	Fehler-Kennung (H)	EA07
8	Fehler-Kennung (M)	Fehler-Kennung (L)	EA08
9	reserviert	Akt. Ident.-Nr.	EA09
10	Basisadresse der Eingangssignalformer		EA0A
11	Basisadresse der Ausgangssignalformer		EA0B
12	Basisadresse Prozessabbild der Eingänge		EA0C
13	Basisadresse Prozessabbild der Ausgänge		EA0D
14	Basisadresse Merkerbereich		EA0E
15	Basisadresse Zeitbereich		EA0F
16	Basisadresse Zählerbereich		EA10
17	Basisadresse Bereich Anschaltung		EA11
18	reserviert	AG-SW-Stand	EA12
19	Endadresse des Anwendermodulspeichers		EA13
20	Basisadresse Bereich System		EA14
21	Länge der DB-Adressliste		EA15
22	Länge der SB-Adressliste		EA16
23	Länge der PB-Adressliste		EA17
24	Länge der FB-Adressliste		EA18
25	Länge der OB-Adressliste		EA19
26	Länge der FX-Adressliste		EA1A
27	Länge der DX-Adressliste		EA1B
28	Länge des Adresslisten-DBs (DB 0)		EA1C
29	Steckplatzkennung	CPU-Kennung 2 (Typ)	EA1D
30	Länge der Bausteinkopfinformation		EA1E
31	CPU-Kennung 1	SW-Stand PG-Ansch.	EA1F
32	Basisadresse DX-Adressliste		EA20
33	Basisadresse FX-Adressliste		EA21

Fortsetzung ...

... Fortsetzung

34	Basisadresse DB-Adressliste	EA22
35	Basisadresse SB-Adressliste	EA23
36	Basisadresse PB-Adressliste	EA24
37	Basisadresse FB-Adressliste	EA25
38	Basisadresse OB-Adressliste	EA26
39 bis 54	reserviert	EA27 bis EA36
55	Zähler für 1 Std. (bis 3599s hex)	EA37
56 bis 59	reserviert für Hantierungsbaustein	EA38 bis EA3B
60 bis 63	reserviert für Anwenderzwecke	EA3C bis EA3F
64 bis 79	reserviert für Systemprogramm	EA40 bis EA4F
80	zusätzliche Fehlererkennung, wenn Bit FE-5 in BS8 gesetzt	EA50
81 bis 127	reserviert für Systemprogramm	EA51 bis EA7F
128	reserviert	EA80
129	reserviert	EA81
130	Kennung Regelung	EA82
131	Anzeigenwort "Alarme gemeinsam sperren"	EA83
132	Anzeigenwort "Alarme gemeinsam verzögern"	EA84
133	Kennung Prozessabbildaktualisierung	EA85
134	reserviert	EA86
135	Anzeigenwort "Weckalarme einzeln sperren"	EA87
136	reserviert	EA88
137	Anzeigenwort "Weckalarme einzeln verzögern"	EA89
138	reserviert	EA8A
139	reserviert	EA8B
140	Anzeigewort "Bausteine schreiben und löschen"	EA8C
141 bis 143	reserviert	EA8D bis EA8F
144	alternatives Laden von Datenbausteinen	EA90
145 bis 255	reserviert	EA91 bis EAFF

Bitbelegung der Systemdaten-wörter

Systemdatum BS0 Unterbrechungsanzeigenwort (EA00h)

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAU	PEU	BAU	MP-STP	ZYK	QVZ	ADF	STP	BCF	FE-3	LZF	REG	STUEB	STUEU	WECK	DOPP

Das Systemdatum BS0 entspricht der STOERUNGSURSACHE im USTACK. Tritt bei der Programmbearbeitung z.B. ein Laufzeitfehler auf, wird das Bit 5 gesetzt. Ist die Programmbearbeitungsebene LZF vollständig bearbeitet, wird das Bit 5 zurückgesetzt.

Systemdatum BS1 Unterbrechungsanzeigen-Löschwort UALW (EA01h)

Aktive Schnittstelle, für Anwender freigegeben!

Durch Setzen von Bit 9 bzw. 10 des UALW erreichen Sie, dass der nächstfolgende ADF bzw. QVZ ignoriert wird und die laufende Programmbearbeitung dadurch nicht beeinflusst wird. Nach Auftreten eines QVZ bzw. ADF setzt das Systemprogramm das betreffende Bit zurück.

Jede Programmbearbeitungsebene hat ihr eigenes UALW.

Belegung BS1:

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	QVZ	ADF	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt

Beispiel zum UALW

In diesem Beispiel wird getestet, ob unter einer bestimmten Peripherieadresse eine Baugruppe ansprechbar ist. Ist die Baugruppe nicht vorhanden, wird mit Hilfe des UALW ein Quittungsverzug verhindert und ein für diesen Fall vorgesehenes Programm bearbeitet.

Ebenso wird getestet, ob eine bestimmte Peripherieadresse im DB 1 eingetragen ist. Falls nicht, wird mit Hilfe des UALW ein Adressierfehler verhindert und ein spezielles Programm bearbeitet.

```
FB 201
NAME   :L
       :SPA FB 10
NAME   :PERITEST     Testen, ob unter der
                     Peripherieadresse 128
PADR   :PB 128       eine Baugruppe ansprechbar ist
MASK   :KM 00000100 00000000
       :SPN =M001
       :..           Dieser Programmteil wird bearbeitet,
                     falls
       :..           Baugruppe nicht ansprechbar ist
       :..
M001   :
       :SPA FB 10
NAME   :PERITEST     Testen, ob im DB 1 eine Baugruppe
PADR   :AB 4         mit der Peripherieadresse 4
MASK   :KM 00000010 00000000     eingetragen ist
       :SPN =M002
       :..      Dieser Programmteil wird bearbeitet,
       :..      falls Peripherieadresse nicht
       :..      eingetragen ist
M002   :
       :BE

FB 10
NAME   :PERITEST
BEZ    :PADR E/A/D/B/T/Z: E BI/BY/W/D: BY
BEZ    :MASK E/A/D/B/T/Z: D KM/KH/KY/KC/KF/KT/KZ/KG: KM
       :L BS 1       UALW laden und sichern
       :T BS 60
       :LW =MASK     QVZ- bzw. ADF-Bit setzen
       :OW
       :T BS 1       UALW zurueckschreiben
       :L =PADR      Einzel-Peripheriezugriff bzw. Zugriff
                     auf
       :             das Prozessabbild
       :L BS 1
       :LW =MASK     QVZ- bzw. ADF-Bit maskieren
       :UW
       :L BS 60      Altes UALW zurueckschreiben, damit
                     der
       :T BS 1       nächste QVZ bzw. ADF erkannt wird
       :TAK
       :BE
```

Systemdatum BS2

Unterbrechungsanzeigen-Sammelwort UAMK (EA02h)

Die 16 Bits des UAMK entsprechen den unter STOERUNGSURSACHE im USTACK aufgeführten möglichen Fehlerursachen. Bei Auftreten eines bestimmten Fehlers wird das zugehörige Bit gesetzt.

Belegung BS2

High-Byte								Low-Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAU	PEU	BAU	MP-STP	ZYK	QVZ	ADF	STP	BCF	FE-3	LZF	REG	STUEB	STUEU	WECK	DOPP

Hinweis!

Das UAMK darf nur gelesen werden!

Beispiel zum UAMK

Geht die CPU aufgrund eines Adressierfehlers (ADF) in STOP, wird im UAMK Bit 9 gesetzt. Wenn nun bei der Bearbeitung des ADF ein Befehlscodefehler (BCF) auftritt, wird im UAMK außerdem Bit 7 gesetzt.

Inhalt des UAMK (binär):	00000010 10000000
Darstellung (hexadezimal) im USTACK:	0280

Während im USTACK unter STOERUNGSURSACHE nur der jeweils zuletzt aufgetretene Fehler angekreuzt ist, sind im UAMK alle bis dahin aufgetretenen Fehler aufsummiert (USTACK Tiefe 05: im UAMK sind 5 Bits gesetzt).

Durch Umwandlung des Hexadezimalcodes in den Binärcode lässt sich der Inhalt des UAMK auswerten. Auf diese Weise können Sie feststellen, welche Fehler den Stoppzustand verursacht haben.

Die Fehler-Bits werden rückgesetzt, sobald die entsprechende Fehler-Programmbearbeitungsebene vollständig bearbeitet ist und damit verlassen wird.

Unterbrechungsanzeigen von Fehlern, denen keine Programmbearbeitungsebene zugeordnet ist (z.B. NAU, PEU, STUEB usw.), werden im ANLAUF gelöscht.

Systemdatum BS5

STOP- und ANLAUF-Kennung (EA05h)

Die Kennungen entsprechen den Steuerbits in Zeile 1 und 2 des USTACK

Belegung BS5

High-Byte: STOP-Kennungen								Low-Byte: ANLAUF-Kennungen							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI-STP	nicht belegt	FE-STP	BEARB-END	PG-STP	STP-SCH	STP-BEF	MP-STP	ANL	nicht belegt	NEUST	MWA	AWA	nicht belegt	NEU-ZUL	MWA-ZUL

Systemdatum BS6 **ZYKLUS- und Modul-/MPL-Kennung (EA06h)**

Die Kennungen entsprechen den Steuerbits in Zeile 3 und 4 des USTACK

Belegung BS6

High-Byte: ZYKLUS-Kennungen								Low-Byte: Modul-/MPL-Kennungen							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RUN	nicht belegt	EIN-PROZ	BARB	OB1-GEL	FB0-GEL	OB-PROZA	OB-WECKA	32KW-RAM	16KW-RAM	8KW-RAM	EPROM	KM-AUS	KM-EIN	DIG-EIN	DIG-AUS

BS7 **URLÖSCH-Kennungen/Fehlerkennungen Initialisieren (EA07h)**

Die Kennungen entsprechen den Steuerbits in Zeile 5 und 6 des USTACK.

Belegung BS7

High-Byte: URLÖSCH-Kennungen								Low-Byte: Fehlerkennungen initialisieren							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
URGELOE	URL-IA	STP-VER	ANL-ABB	UA-PG	UA-SYS	UA-PRFE	UA-SCH	DX0-FE	nicht belegt	MOD-FE	RAM-FE	DB0-FE	DB1-FE	DB2-FE	KOR-FE

BS8 **Fehlerkennungen HW/SW (EA08h)**

Die Kennungen entsprechen den Steuerbits in Zeile 7 und 8 des USTACK.

Belegung BS8

High-Byte: Fehlerkennungen HW								Low-Byte: Fehlerkennungen SW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAU	PEU	BAU	STUE-FE	ZYK	QVZ	ADF	WECK-FE	BCF	nicht belegt	FE-5	power-down-Fehler	FE-3	LZF	REG-FE	DOPP-FE

BS29 **Steckplatzkennung/CPU-/AG-Typ (EA1Dh)**

Belegung BS29

High-Byte: Steckplatzkennung CPU								Low-Byte: Steckplatzkennung AG-Typ							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nicht belegt	nicht belegt	nicht belegt	nicht belegt	CPU-Nr.4	CPU-Nr.3	CPU-Nr.2	CPU-Nr.1	CPU-Typ	CPU-Typ	CPU-Typ	CPU-Typ	AG-Typ	AG-Typ	AG-Typ	AG-Typ

High-Byte: Aktive Schnittstelle, wird von den Hantierungsbausteinen sowie vom OB 218 und den Operationen SES und SEF benutzt.

Low-Byte: CPU-Typ: 1011 CPU 928B

1011 CPU 928C Aus Kompatibilitätsgründen gleich

AG-Typ: 0111 AG 135U

BS64 **Speicher-Füllstandsanzeiger** des Anwenderspeichers
CPU 928B: (0000h ... 7FFFh)
CPU 928C: (0000h ... FFFFh)

BS65 **Speicher-Füllstandsanzeiger** des DB-RAMs
CPU 928B: (8000h ... DD7Fh)
CPU 928C: (0000h ... DD7Fh)

BS80 **Fehlerspezifikation (EA50h)**

Hier finden Sie zusätzliche Informationen zur Spezifizierung des aufgetretenen Fehlers, wenn Bit 5 in BS8 gesetzt bzw. FE-5 im USTACK angekreuzt ist.

Kennung in BS80	Fehlerursache
2460h	Ready-Signal auf Bus steht dauernd an

Fehlerursache	BS80	LED		
		QVZ	ADF	ZYK
Hardware Watchdog	FE01H	O	O	X
CP-SPS Kommunikation	FE02H	O	X	O
Überwachung SPS-Prozessor	FE03H	O	X	X
Tasküberwachung CP-Prozesssor	FE04H	X	O	O
Nicht belegter Interrupt	FE05H	X	O	X
Speicherfehler	FE06H	X	X	O

O => LED aus; X => LED blinkt

BS131, **Anzeigenwort "Alarme gemeinsam sperren" (EA83h)**

Das Systemdatum BS131 zeigt Ihnen folgende Zustände der Programmbearbeitungsebenen "Alarmbearbeitung" an:

Belegung BS131

High-Byte								Low-Byte: Alarme gemeinsam sperren							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Verzögerungs-alarm	Prozess-alarm	uhrzeitgesteuerter Weckalarm	Weckalarme mit festem Zeitraster

Bit = 1 bedeutet, dass Alarm gesperrt ist.

BS132 **Anzeigenwort "Alarme gemeinsam verzögern" (EA84h)**

Das Systemdatum BS132 zeigt Ihnen folgende Zustände der Programmbearbeitungsebenen "Alarmbearbeitung" an:

Belegung BS132

High-Byte								Low-Byte: Alarme gemeinsam verzögern							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	Verzögerungs-alarm	Prozess-alarm	uhrzeitgesteuerter Weckalarm	Weckalarme mit festem Zeitraster

Bit = 1 bedeutet, dass Alarm verzögert ist.

BS133 **Prozessabbildaktualisierung (EA85h)**

Jedes Bit verhindert, falls es gesetzt ist, die Prozessabbildaktualisierung einmal, anschließend wird es vom Systemprogramm wieder auf '0' gesetzt.

Belegung BS133:

High-Byte								Low-Byte: Prozessabbildaktualisierung							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	KM-AUS	KM-EIN	DIG-EIN	DIG-AUS

Bit 0 = 0 Nächstes Prozessabbild der digitalen Ausgänge wird ausgegeben.

Bit 0 = 1 Nächstes Prozessabbild der digitalen Ausgänge wird unterdrückt.

Bit 1 = 0 Nächstes Prozessabbild der digitalen Eingänge wird eingelesen.

Bit 1 = 1 Nächstes Prozessabbild der digitalen Eingänge wird unterdrückt.

BS135 **Anzeigenwort "Weckalarme einzeln sperren" (EA87h)**

Das Systemdatum BS135 zeigt folgende Zustände der Programmbearbeitungsebenen "Weckalarmbearbeitung" an:

Belegung BS135:

High-Byte: Weckalarme einzeln sperren								Low-Byte: Weckalarme einzeln sperren							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Weckal. 5s OB18	Weckal. 2s OB17	Weckal. 1s OB16	Weckal. 500ms OB15	Weckal. 200ms OB14	Weckal. 100ms OB13	Weckal. 50ms OB12	Weckal. 20ms OB11	Weckal. 10ms OB10	0	0	0

Bit = 1 bedeutet, dass Weckalarm gesperrt ist.

BS137 **Anzeigenwort "Weckalarme einzeln verzögern" (EA89h)**

Das Systemdatum BS137 zeigt folgende Zustände der Programmbearbeitungsebenen "Weckalarmbearbeitung" an:

Belegung BS137:

High-Byte: Weckalarme einzeln verzögern								Low-Byte: Weckalarme einzeln verzögern							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Weckal. 5s OB18	Weckal. 2s OB17	Weckal. 1s OB16	Weckal. 500ms OB15	Weckal. 200ms OB14	Weckal. 100ms OB13	Weckal. 50ms OB12	Weckal. 20ms OB11	Weckal. 10ms OB10	0	0	0

Bit = 1 bedeutet, dass Weckalarm verzögert ist.

BS138 **Schreibschutz für Anwenderspeicher im EPROM-Betrieb**

Adresse EA8Ah

Der Anwenderspeicher ist nach dem Kopieren der Bausteine von einer Memory Card (EPROM-Betrieb) schreibgeschützt. Bausteine mit EPROM-Kennung können vom PG nicht überschrieben werden, jedoch vom Anwenderprogramm!

Durch Löschen oder Setzen der **Schreibschutzkennung Bit 0** in BS 138 über das PG und anschließenden NEUSTART kann der Schreibschutz gezielt aus- und wieder eingeschaltet werden. (Die Bits Nr. 1 bis 15 von BS 138 sind ohne Bedeutung.)

Vorgehen beim Ein-/Ausschalten der Schreibschutzkennung

1. Lassen Sie sich den Inhalt der Adresse **EA8Ah** am PG ausgeben.
2. Setzen bzw. Löschen Sie das Bit 0 von BS 138 durch Überschreiben des Inhalts der Adresse **EA8Ah** mit dem Bitmuster 000xh (x=1 für "Setzen", x=0 für „Löschen“ der Schreibschutzkennung).
3. Führen Sie anschließend einen NEUSTART durch. Nach der Bearbeitung des OB 20 wird das Bit 0 im Systemdatenwort BS 138 ausgewertet und der Schreibschutz entsprechend aktiviert oder deaktiviert.

BS 138 kann nicht nur per PG sondern auch per OB 20 gesetzt werden.

Der Status der Schreibschutzkennung bleibt bis zum nächsten URLÖSCHEN der CPU erhalten.

CPU 928C

Das Kopieren des Anwenderprogramms in das interne Flash-EPROM über BS138 ist bei der CPU 928C nicht möglich.

BS140 **Anzeigenwort "Bausteine schreiben und löschen" (EA8Ch)**

Das Systemdatum BS140 zeigt Ihnen an, ob seit dem letzten URLÖSCHEN der CPU bzw. seit dem letzten Löschen des Systemdatums BS140 Bausteine überschrieben, geladen oder gelöscht wurden.

Für jeden Baustein werden Bits für Veränderung und Bausteintyp als Oder-Verknüpfung hinzugefügt.

Vor einem neuen Überwachungsabschnitt muss das Systemdatum BS140 gelöscht werden.

URLÖSCHEN löscht BS140.

Belegung BS140:

High-Byte: Schreib-/Lese-Kennungen								Low-Byte: Schreib-/Lese-Kennungen							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baustein gelöscht	Baustein geladen	Baustein über-schrieben	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	nicht belegt	DX	DB	FX	FB	SB	PB	OB

BS144 **Anzeigenwort "Alternatives Laden von Datenbausteinen in das DB-RAM" (EA90h)**

Standardmäßig werden bei der CPU 928B alle Bausteine vom PG zuerst in das Anwenderspeicher-Modul geladen. Erst wenn dort kein Platz mehr ist, werden Datenbausteine und nur diese im DB-RAM abgelegt.

Über das Bit 0 von BS144 können Sie die Reihenfolge beim Laden der Datenbausteine beeinflussen:

Bit 0 =	0	Voreinstellung "Standard-Verhalten" Die DBs werden zuerst in das Anwender-speicher-Modul geladen. Sobald dieser voll ist, werden die DBs im DB-RAM abgelegt.
Bit 0 =	1	Die DBs werden zuerst in das DB-RAM geladen. Erst wenn dort kein Platz mehr ist, werden sie in das Anwenderspeicher-Modul übernommen.

Die übrigen Bits von BS144 sind nicht belegt.

Hinweis!

Codebausteine werden unabhängig von der Einstellung in BS144 **nur in den Anwenderspeicher** geladen. Die Einstellung in BS144 hat keinen Einfluss auf Operationen und Sonderfunktions-OBs zum Erzeugen und Umladen von Bausteinen.

928C

Das Alternative Laden von Datenbausteinen ist bei der CPU 928C nicht erforderlich, da Datenbausteine immer im DB-RAM abgelegt werden.

Teil 9 Speicherzugriff über absolute Adressen

Überblick

In diesem Kapitel erfahren Sie, wie Sie mit Hilfe von Operationen und speziellen Registern Daten in bestimmten Speicherbereichen über Absolutadressen ansprechen können.

Bei den Operationen mit dem Basisadressregister werden die Zugriffe auf den lokalen -, globalen Speicher sowie auf den Kachelspeicher beschrieben.

Nachfolgend sind beschrieben:

- LIR/TIR: 16-bit-Register
- Beispiele für die Übertragung der Speicherblöcke
- Transferoperationen zwischen den Registern
- Zugriffe auf den lokalen Speicher
- Zugriffe auf den globalen Speicher
- Zugriffe auf den Kachelspeicher

Inhalt

Thema **Seite**

Teil 9 Speicherzugriff über absolute Adressen.....9-1

Einführung 9-2

Speicherzugriffe über Adresse in AKKU 1 9-6

Speicherblöcke transferieren 9-15

Operationen mit dem Basisadressregister (BR-Register) 9-22

Einführung

Die Programmiersprache STEP® 5 von Siemens enthält Operationen, mit denen Zugriffe auf den gesamten Adressraum möglich sind. Diese Operationen gehören zu den "Systemoperationen".

Achtung!

Bei einer nicht sachgerechten Anwendung dieser Operationen können Bausteine und Systemdaten überschrieben werden. Dies kann unerwünschte Betriebszustände zur Folge haben. Operationen, die mit absoluten Adressen arbeiten, sollten deshalb nur von Anwendern mit sehr guten Systemkenntnissen benutzt werden.

Lokaler Speicher

Als lokaler Speicher wird der Speicherbereich bezeichnet, der auf jeder CPU vorhanden ist (Anwenderspeicher, DB-RAM, BA-, BB-, BS-, BT-Bereich, Zähler, Zeiten, Merker, Prozessabbild).

Globaler Speicher

Der globale Speicher ist nur einmal für alle CPUs vorhanden und wird über den Rückwand-Bus adressiert.

Speicher-organisation

Speicherbereiche sind byte- oder wortweise organisiert.

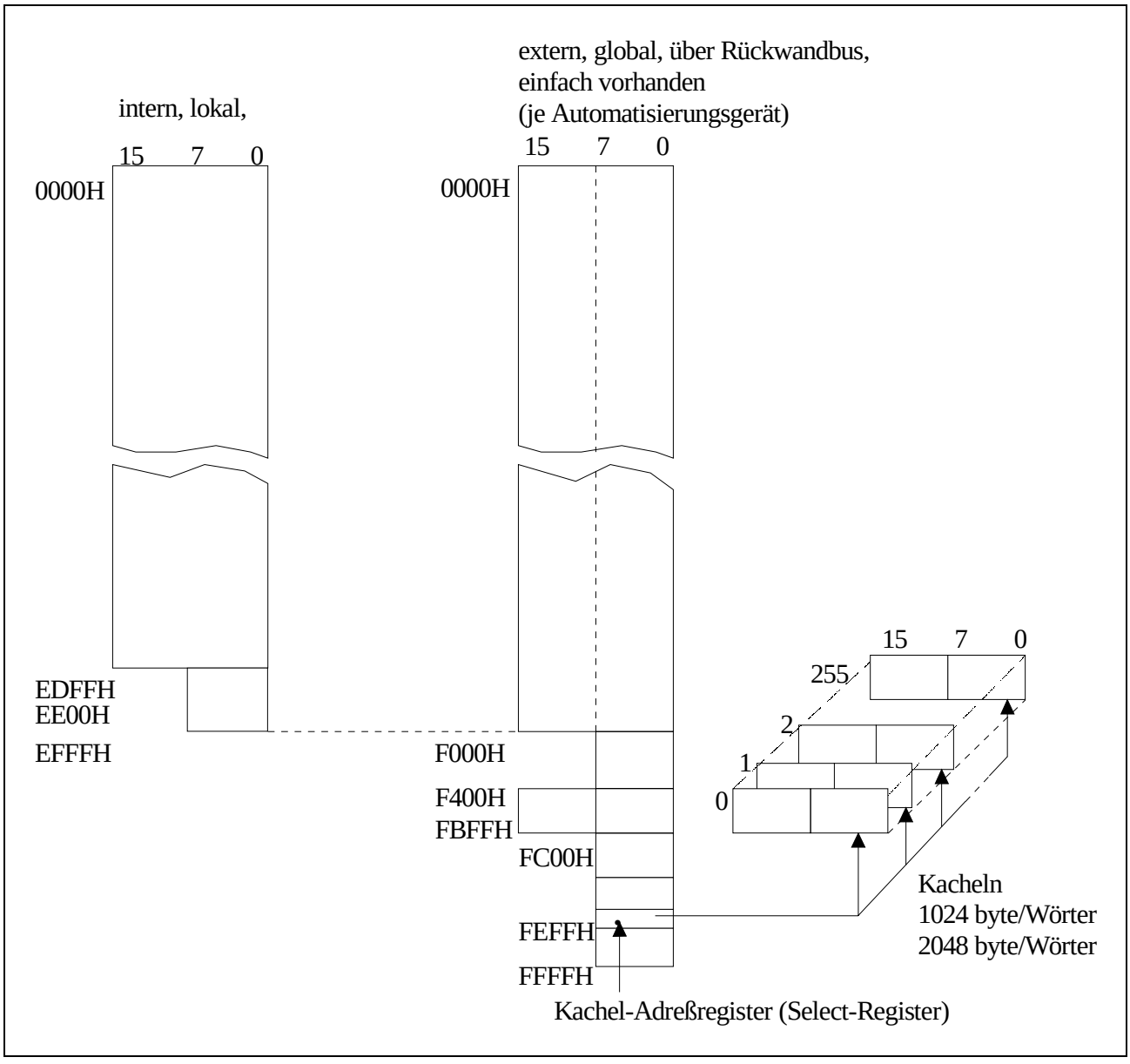
Byteweise Organisation

Jede Adresse adressiert ein Byte.

Wortweise Organisation

Jede Adresse adressiert ein 16-bit-Wort (2 Bytes).

Globaler und lokaler Speicher



Speicherzugriffe

Die folgenden Operationen ermöglichen Zugriffe auf lokale bzw. globale Speicherbereiche über absolute Adressen.

Zugriffe auf den Lokal- und Globalbereich

Sie können sowohl auf den Lokal- als auch auf den Globalbereich zugreifen:

Lokalbereich
(Adressen 0000h bis EFFFh) mit:

LIR, TIR, TNB, TNW

byteweise organisierter Teil des Globalbereichs
(Adressen F000h bis F3FFh, FC00h bis FFFFh) mit:

LIR, TIR, TNB, TNW,

auf den wortweise organisierten Teil des Lokalbereiches
(Adressen 0000h bis E3FFh sowie E800h bis EDFFh) mit:

LRW, TRW, LRD, TRD.

Zugriffe auf den Kachelbereich

Sie können auf folgende Teile des Kachelbereichs zugreifen:

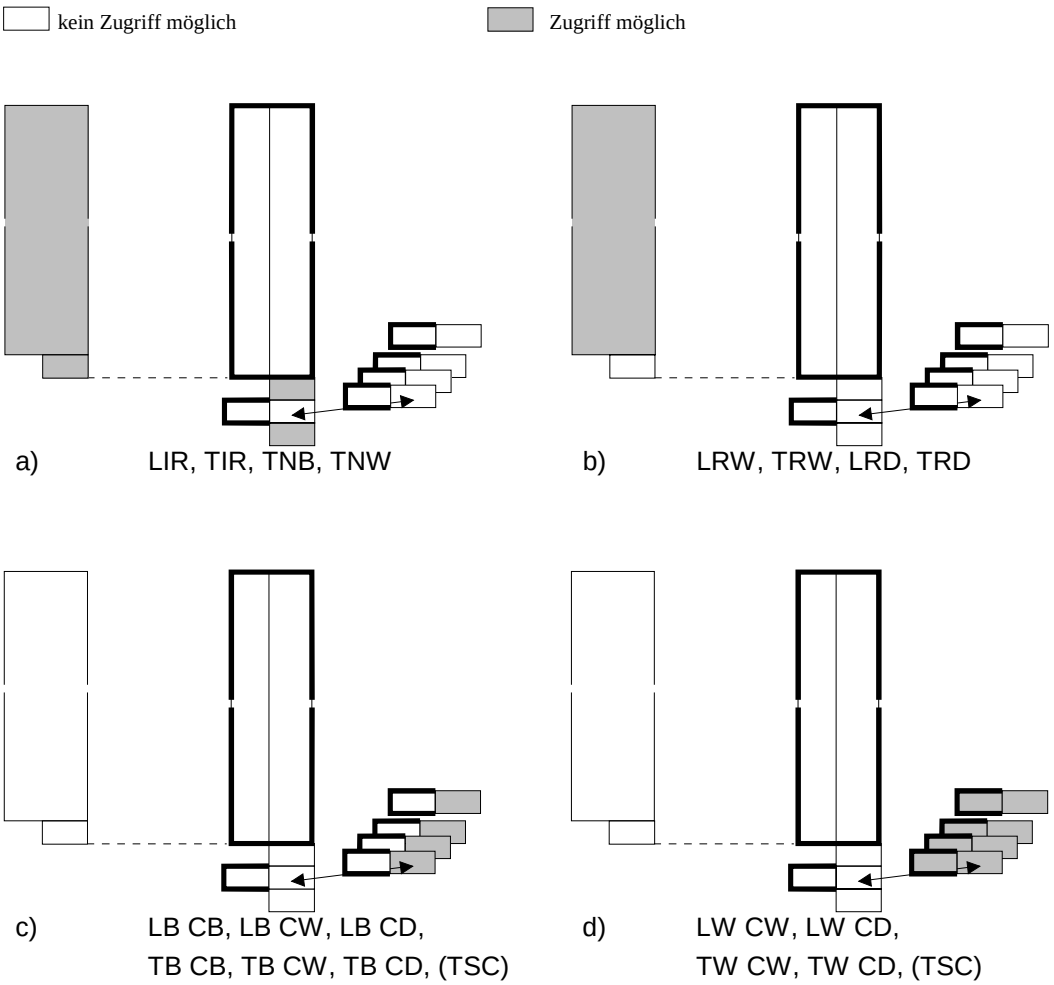
auf den byteweise organisierten Teil des Globalbereichs
(Adressen F400h bis FBFFh, = Kachelbereich):

LB CB, LB CW, LB CD, TB CB, TB CW, TB CD, TSC

auf den wortweise organisierten Teil des Globalbereichs
(Adressen F400h bis FBFFh, = Kachelbereich):

LW CW, LW CD, TW CW, TW CD, TSC

Zugriffe auf lokale bzw. globale Speicherbereiche über absolute Adressen



Speicherzugriffe über Adresse in AKKU 1

Hinweis!

Mit den Befehlen LIR und TIR ist bei der CPU 928C kein Zugriff auf den Anwenderspeicher möglich! Verwenden Sie hier die Befehle LDI und TDI.

Die Register sind Betriebsmittel der CPU zum Bearbeiten des Programms. Jedes Register ist 16 Bit breit. Mit den Systemoperationen LIR (Lade indirekt Register) und TIR (Transferiere indirekt Register) können Sie auf die Inhalte der Register zugreifen.

Operationen

Operation	Operand	Funktion
LIR	Register-Nr. 0 bis 15	Das angegebene Register wird mit dem Inhalt einer Speicherzelle (durch AKKU1 adressiert) geladen.
TIR	Register-Nr. 0 bis 15	Der Inhalt des angegebenen Registers wird in eine Speicherzelle (durch AKKU1 adressiert) transferiert.

Das Speicherwort liegt entweder im Lokalbereich (Adressen 0000h bis EFFFh) oder im byteweise organisierten Teil des Globalbereichs (Adressen F000h bis F3FFh, FC00h bis FFFFh).

Operationen

Operation	Operand	Funktion
LDI	A1 A2	Das angegebene Register wird mit dem Inhalt einer Speicherzelle (durch AKKU1 adressiert) geladen.
TDI	A1 A2	Der Inhalt des angegebenen Registers wird in eine Speicherzelle (durch AKKU1 adressiert) transferiert.

Auf den folgenden Seiten erfahren Sie, welche Register Sie bei den Operationen verwenden können.

LIR/TIR 16-bit-Register indirekt laden/ transferieren

Die folgende Tabelle zeigt Ihnen, welche Registernummern Sie bei der CPU 928 für die Operationen LIR und TIR verwenden dürfen und wie diese belegt sind.

Register-Nr.	Registerbelegung (je 16 bit breit)	
0	AKKU-1-H	(linkes Wort von AKKU1, Bit 16-31)
1	AKKU-1-L	(rechtes Wort von AKKU1, Bit 0-15)[1)]
2	AKKU-2-H	
3	AKKU-2-L	
6	DBA	(Datenbaustein-Anfangsadresse)
8	DBL	(Datenbaustein-Länge)
9	AKKU-3-H	
10	AKKU-3-L	
11	AKKU-4-H	
12	AKKU-4-L	
15	SAZ	(STEP-Adresszähler)

1) Soll über AKKU-1-L der Inhalt der adressierten Speicherzelle in das Register '1' geladen werden, so wird dadurch die in AKKU-1-L hinterlegte Adresse überschrieben.

Die Register 4, 5, 7, 13 und 14 sind bei der CPU 928 nicht vorhanden. LIR/TIR-Operationen mit diesen Registernummern werden wie eine Null-operation (NOP) behandelt.

LIR und TIR auf Kachelbereich

Die Befehle LIR und TIR sind für Automatisierungsgerät AG-135U für den Zugriff auf den Kachelbereich (Adressen F400h bis FBFFh) nicht zu empfehlen. Verwenden Sie stattdessen die Befehle aus "Zugriff auf den Kachelspeicher" oder die Sonderfunktionen (OB 216 bis OB 218) für "Kachelzugriffe".

Hinweis!

Die dazugehörigen Lade und Transferoperationen (LB CB, TB CB) für byteweise organisierte Kacheln finden Sie in diesem Kapitel.

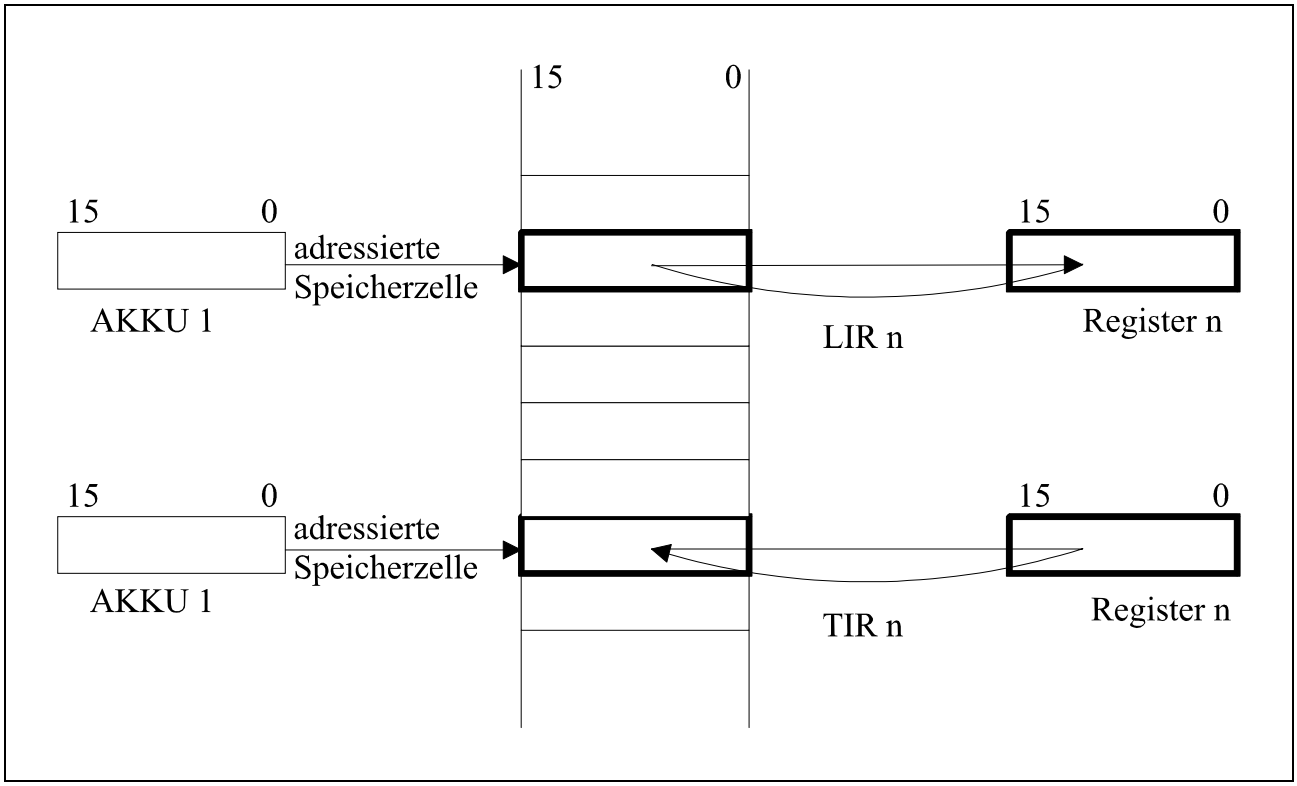
LIR/TIR auf 8-bit-Speicherbereiche

Wird mit LIR/TIR auf Speicherbereiche zugegriffen, die nur 8 bit breit sind (Speicheradressen von E400h bis E7FFh sowie ≥ EE00h), so beachten Sie, dass

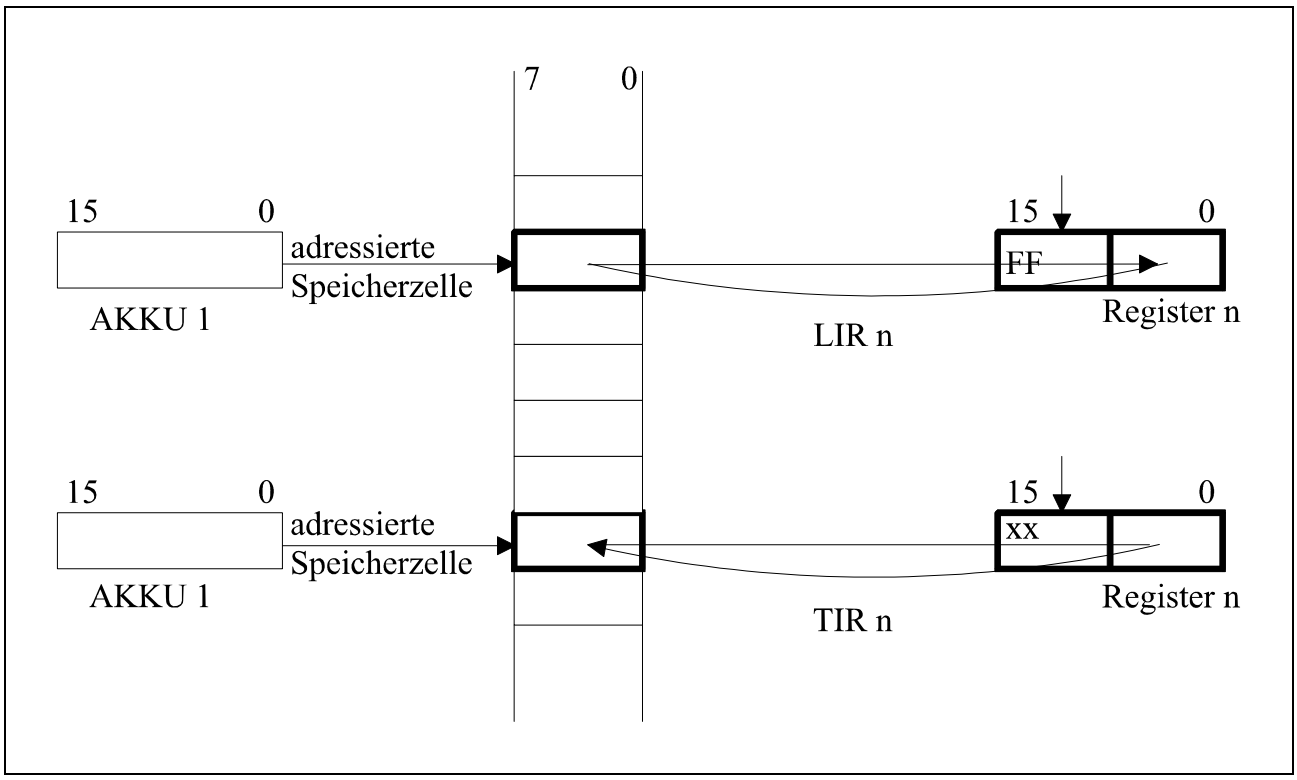
- bei TIR nur das Low-Byte des Registers übertragen wird (das High-Byte des Registers geht verloren)
- bei LIR das High-Byte des Registers mit FFh überschrieben wird.

Die folgenden Abbildungen zeigen Ihnen den Unterschied bei Zugriffen mit LIR/TIR auf wort- und byteweise orientierte Speicherbereiche:

LIR/TIR auf 16-bit Speicherbereiche (wortweise orientiert)



LIR/TIR auf 8-bit-Speiserbereiche (byteweise orientiert)



Register 0 bis 3 und 9 bis 12: AKKU 1, 2, 3 und 4

Die Akkumulatoren werden von der CPU bei der Programmbearbeitung als Zwischenspeicher verwendet. Mit den Operationen TIR und LIR können Sie die Inhalte der AKKUs in absolut adressierte Speicherzellen transferieren bzw. die Inhalte absolut adressierter Speicherzellen in die AKKUs laden. Die Absolutadresse der Speicherzelle steht jeweils im AKKU-1-L..

Beispiele

Der Inhalt der Speicherzelle mit der Adresse A000 wird ins Merkerwort MW100 geladen.

```
:L     KH A000    Adresse A000 der Speicherzelle in AKKU
                  1 laden
:LIR   1          Inhalt der durch AKKU 1 adressierten
:                 Speicherzelle ins Register 1 = AKKU-1-
                  L laden
:T     MW 100     Inhalt der Adresse A000 im Merkerwort
                  MW 100 ablegen
:BE
```

Der Inhalt des Merkerwortes 200 wird in die Speicherzelle mit der Adresse A000 transferiert.

```
:L     MW 200     Merkerwort MW 200 in den AKKU 1 laden
:L     KH A000    Adresse A000, auf die transferiert
:                 werden soll, in den AKKU 1 laden
                  (Merkerwort MW 200 nach AKKU 2)
:TIR   3          Inhalt von Register 3 = AKKU-2-L in
:                 die durch AKKU 1 adressierte
                  Speicherzelle transferieren
:BE
```

Register 6: DBA (Datenbaustein-Anfangsadresse)

Beim Aufschlagen eines Datenbausteins mit den Befehlen A DB und AX DX wird das Register 6 mit der Adresse des DW 0 im aufgeschlagenen Datenbaustein geladen. Diese Adresse ist in der Bausteinadressliste im DB 0 enthalten.

Das DBA-Register wird vor jedem Aufruf des OB1 oder FB0 gleich '0' gesetzt.

Das DBA-Register bleibt erhalten, wenn

- durch eine Sprunganweisung (SPA/SPB) die Programmbearbeitung in einem anderen Baustein fortgesetzt wird

  oder

- eine andere Programmbearbeitungsebene eingeschachtelt wird.

Es ändert sich, wenn

- ein anderer Datenbaustein aufgeschlagen wird

  oder

- ein Rücksprung in einen übergeordneten Baustein erfolgt, nachdem im aufgerufenen Baustein ein neuer Datenbaustein aufgeschlagen wurde.

Hinweis!

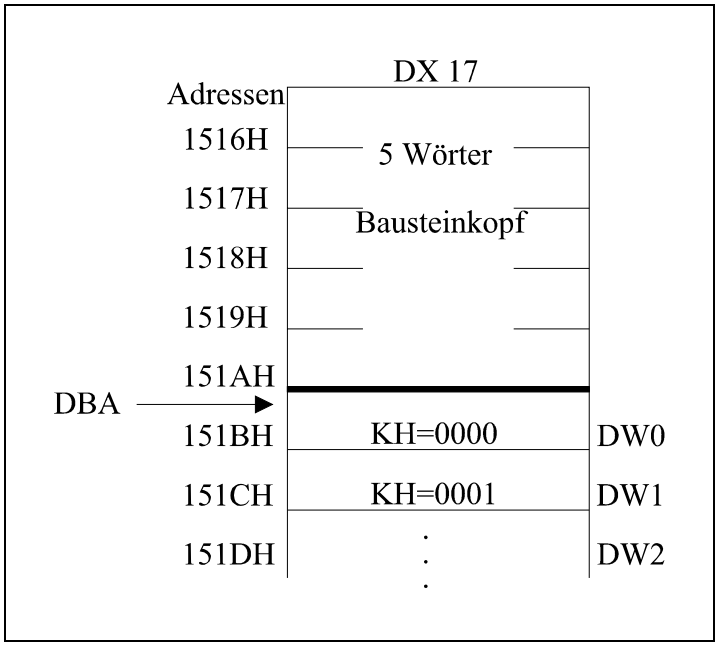
Im USTACK ist die im DBA-Register eingetragene Adresse unter 'DB-ADR' angegeben.

Zugriffe auf Datenwörter erfolgen normalerweise mit den Operationen L/T DW, L/T DR, L/T DL, L/T DD, U/O/UN/ON/=/S/R Dx.y. Diese sind jedoch nur bis Datenwort DW 255 möglich. Durch Manipulieren des DBA-Registers können Sie mit diesen Befehlen auch auf Datenwörter > 255 zugreifen. Sie können dies auch mit Hilfe des Sonderfunktions-Organisationsbausteins OB180 erreichen.

Beispiele

Beispiel 1:

Wirkung der Operation "AX DX 17" auf das DBA-Register:



Bei Aufruf des DX 17 wird im DBA-Register die Adresse des Speicherwortes, in welchem das Datenwort DW<N>0 hinterlegt ist, im Beispiel: DBA = 151BH, eingetragen.

Hinweis: Im USTACK ist die im DBA-Register eingetragene Adresse unter 'DB-ADR' angegeben.

Beispiel 2:

Durch Ändern von Register 6 wird das Datenwort DW 300 des DB 100 geladen.

```
        FB7
NAME    :LIR/TIR6
        :L      BS 34     Anfangsadresse der DB-Adressliste plus
                          100
        :ADD    BF +100   ergibt Adresslisteneintrag des DB 100
        :LIR    1         Anfangsadresse des DB 100 (DW 0) nach
                          AKKU1
        :ADD    KF +200   Adresse des DW 200 im DB 100 im
        :T      BS 62     Systemdatenwort BS 62 ablegen
        :L      BS 20     Basisadresse Systemdaten laden
        :ADD    KF +62    Adresse des BS 62 in AKKU 1 laden
        :LIR    6         DBA-Register mit dem Inhalt der Adr.
        :                 des BS 62 laden, d.h. der Datenbau-
                          steinanfang wird auf DW 200 gesetzt.
        :L      DW 100    DW (200+100) = DW 300 laden
        :T      MW 100    DW 300 im Merkerwort MW 100 ablegen
        :BE
```

Beispiel 3:

Verändern von DBA- und DBL-Register

```
        FB7
NAME    OB180
        :A      DB 100    DBA- und DBL-Register mit den Werten
        :L      KF 200    des DB 100
        :SPA    OB 180    laden und mit Hilfe des OB180 das DBA-
                          Register um 200 erhöhen und das DBL-
                          Register um 200 vermindern
        :SPB    =FEHL     Fehlerausgang, falls der DB 100 weniger
        :                 oder gleich 200 Datenwörter enthält
        :L      DW 100    DW 300 laden und
        :T      MW 100    im MW 100 ablegen
        :BEA
FEHL    :                 Programmteil zur Fehlerbehandlung
        :
        :BE
```

Hinweis!

Wenn Sie wie im Beispiel 2 das DBA-Register verstellen, wird das DBL-Register nicht verändert. Damit ist eine Transferfehlerüberwachung nicht mehr gewährleistet! Durch Anwendung des Sonderfunktions-OBs 180 "Variabler Datenbausteinzugriff" können Sie das DBA-Register ebenfalls um eine vorgegebene Anzahl an Datenworten verschieben. Da der OB180 gleichzeitig das DBL-Register verändert, werden Transferfehler weiterhin überwacht.

Register 8: DBL = Daten-baustein-Länge

Zusätzlich zum DBA-Register wird bei jedem Aufruf eines Datenbausteins das DBL-Register geladen. Es enthält die Länge (in Worten) des aufgerufenen Datenbausteins ohne Baustein-Kopf. Das DBL-Register wird vor jedem Aufruf des OB1 oder FB0 gleich '0' gesetzt.

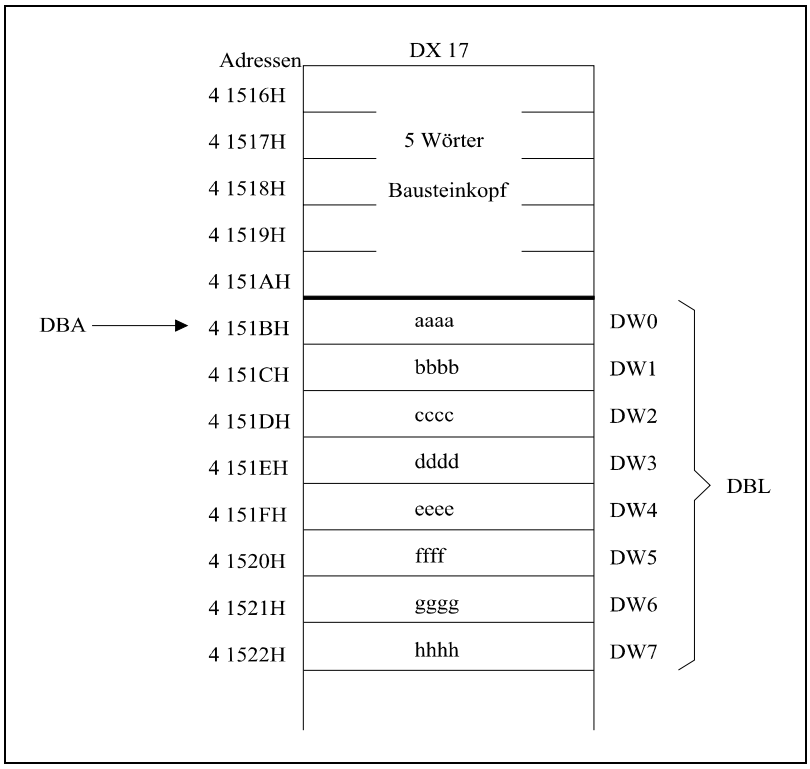
Das DBL-Register bleibt erhalten, wenn:

- durch eine Sprunganweisung (SPA/SPB) die Programmbearbeitung in einem anderen Baustein fortgesetzt wird
  oder
- eine andere Programmbearbeitungsebene eingeschachtelt wird.

Es ändert sich, wenn:

- ein anderer Datenbaustein aufgeschlagen wird
  oder
- ein Rücksprung in einen übergeordneten Baustein erfolgt, nachdem im aufgerufenen Baustein ein neuer Datenbaustein aufgeschlagen wurde.

Wirkung der Operation "AX DX 17" auf das DBL:



Bei Aufruf des DX 17 wird im DBL-Register die Anzahl der vorhandenen Datenwörter eingetragen, im Beispiel: DBL = 8 (DW 0 bis DW 7)

Hinweis: Im USTACK ist die im DBL-Register eingetragene Anzahl unter "DBL-REG" angegeben.

Register 15: SAZ = STEP-Adresszähler

Im Register 15 ist während der Programmbearbeitung die Absolutadresse derjenigen Operation im Programmspeicher eingetragen, die als nächste zu bearbeiten ist.

Beispiele für die Anwendung der Register

Beispiel 1: Alle Datenwörter eines Datenbausteins sollen mit einer Konstanten beschrieben werden. Das unten dargestellte Programm beschreibt alle Datenwörter des DB 50 mit der Konstanten KH = A5A5. Nach Ändern der fettgedruckten Befehle kann es auch zum Beschreiben anderer Datenbausteine (DB oder DX) mit beliebigen Werten benutzt werden. Nicht vorhandene Datenbausteine oder Datenbausteine mit null Datenworten werden erkannt und führen zum Sprung zur Marke NIVO. Die Anfangsadresse (DBA) und Länge (DBL) des Datenbausteins wird über die Sonderfunktion OB181 "Datenbaustein (DB/DX) testen" ermittelt.

Das Programm nutzt alle vier Akkumulatoren. Im Bild sehen Sie die Belegung der Akkumulatoren während des Prozessablaufs bis zur Marke SCHL. Innerhalb der Schleife ändert sich die Akkumulatorbelegung nicht.

Der AKKU 1 enthält zunächst die Adresse des letzten Datenwortes (DBA + DBL -1) und wird mit jedem Schleifendurchlauf um eins vermindert.

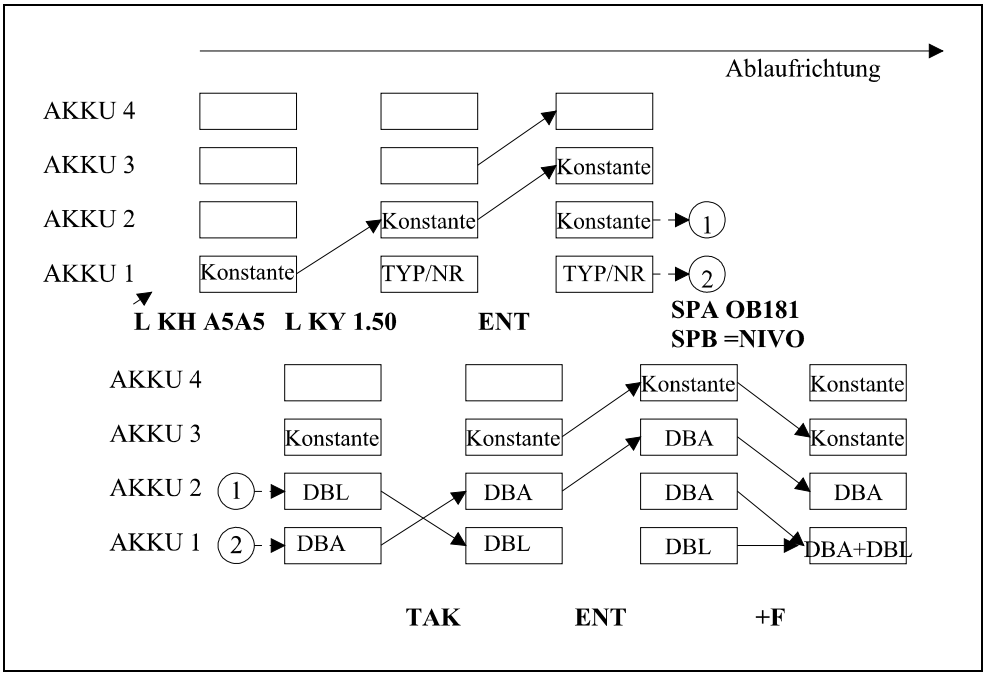
Der AKKU 2 enthält die Adresse des ersten Datenwortes (DBA). Die Schleife wird abgebrochen, sobald der Inhalt des AKKU 1 kleiner als der Inhalt des AKKU 2 ist.

Zum Beschreiben der Datenwörter wird der Befehl TIR 10 verwendet, der den Inhalt des AKKU-3-L (die Konstante) unter der im AKKU-1-L stehenden Adresse abspeichert.

```
        :
        :L      KHA5A5  Konstante, mit der alle DW beschrieben werden
        :               sollen
        :L      KY 1,50 Typ und Nummer des Datenbausteins
        :ENT
        :SPA    OB181   Sonderfunktions-OB "Datenbausteine testen"
        :SPB    =NIVO   Abbruch, falls DB 50 nicht vorhanden
        :TAK
        :ENT
        :+F
        :               AKKU 1: = Adresse des letzten Datenworts +1
        :               AKKU 2: = Adresse des ersten Datenworts
        :               AKKU 3: = Konstante
        :!=F            Abbruch, falls der DB 50 null Datenwörter
                        enthält
        :SPB    =NIVO
        :
SCHL    :ADD    BF-1    Alle Datenwörter, beginnend mit dem letzten DW,
        :TIR    10      mit der im AKKU-3-L enthaltenen Konstanten
                        beschrieben
        :
        :><     F       Abfrage, ob 1. Datenwort erreicht
        :
        :SPB    =SCHL   Rücksprung in die Schleife, wenn 1. Datenwort
        :               <N>  noch nicht erreicht
        :               Fortsetzung des Programms...
WEIT    :               nachdem alle Datenwörter beschrieben wurden
        :
        :BEA
NIVO    :               falls der DB 50 nicht vorhanden ist oder null
        .               Datenwörter hat
        :BE
```

Hinweis: Der Programmteil ab Marke "SCHL" kann zum Beschreiben beliebiger Speicherbereiche (z.B. Merker, Zeiten, Zähler) mit einer Konstanten benutzt werden.

Belegung des AKKUs während des Programmablaufs



Beispiel 2: Löschen aller Merkerbytes (MB 0 bis MB 255)

```
        :L     KB 0        Konstante, mit der alle Merkerbytes
        :                  beschrieben werden sollen
        :L     BS 14       Basisadresse des Merkerbereichs
        :                  (= Adresse des ersten Merkerbytes MB
                           0)
        :ENT
        :L     KF +256     + Länge des Merkerbereichs
        :ENT               = (Adresse des letzten Merkerbytes MB
                           255) +1
        :+F
        :
SCHL    :ADD   BF -1       Alle 256 Merkerbytes, beginnend mit
        :TIR   10          dem Merkerbyte MB 255, mit der im
        :                  AKKU-3-LL enthaltenen Konstanten
                           beschreiben
        :SPB   =SCHL
        :
```

Speicherblöcke transferieren

Hinweis!

Mit den Befehlen LIR und TIR ist bei der CPU 928C kein Zugriff auf den Anwenderspeicher möglich! Verwenden Sie hier die Befehle LDI und TDI.

Anwendung

Mit den Systemoperationen TNB und TNW können Sie Speicherblöcke transferieren (max. 255 Byte mit TNB, max. 255 Worte mit TNW).

Sie können mit diesen Operationen sowohl auf den lokalen Speicherbereich als auch auf den byteweise organisierten Teil des globalen Speicherbereichs (Adressen F000H bis F3FFH, FC00H bis FFFFH) zugreifen.

Operationen

Operation	Operand	Funktion
TNB	0 bis 255	Blocktransfer 0 bis 255 Byte
TNW	0 bis 255	Blocktransfer 0 bis 255 Worte

Parameter

Blocklänge

Operand = Anzahl Bytes (TNB) bzw. Anzahl Worte (TNW)

Endadresse des Quellbereichs

AKKU-2-L = Endadresse des Quellbereichs

Endadresse des Zielbereichs

AKKU-1-L = Endadresse des Zielbereichs

Quellbereich und Zielbereich müssen vollständig in einem der in folgender Tabelle genannten Speicherbereiche liegen und dürfen sich nicht überschneiden.

Erlaubte Speicherbereiche

Adressen	Speicherbereich CPU 928B	Speicherbereich CPU 928C
0000h bis 7FFFh	Anwenderspeicher: Anwender-Modul (16 bit) 32K-Worte	DB-RAM
8000h bis DD7Fh	System-RAM DB-RAM (16 bit)	
DD80h bis E3FFh E400h bis E7FFh E800h bis EDFFh EE00h bis EFFFh F000h bis FFFFh	DB 0 (16 bit) S-Merker (8 bit) Systemdaten (16 bit: BA, BB, BS, BT, Zähler und Zeiten) RAM (8 bit: Merker, Prozessabbild) Peripherie (8 bit)/Rückwand-Bus	

Ablauf

Der Blocktransfer erfolgt dekrementierend, d.h. er beginnt seine Übertragung mit der höchsten Adresse des Quellbereichs (= Endadresse) und beendet sie mit der niedrigsten.

Anwendung auf den Kachelbereich

Die Operationen TNB und TNW sind im Automatisierungsgerät AG 135U nicht für den Zugriff auf den Kachelbereich (Adressen F400H-FBFFH) geeignet. Verwenden Sie stattdessen die Operationen aus "Zugriff auf den Kachelspeicher" oder die Sonderfunktionen aus dem Teil "Kachelzugriffe".

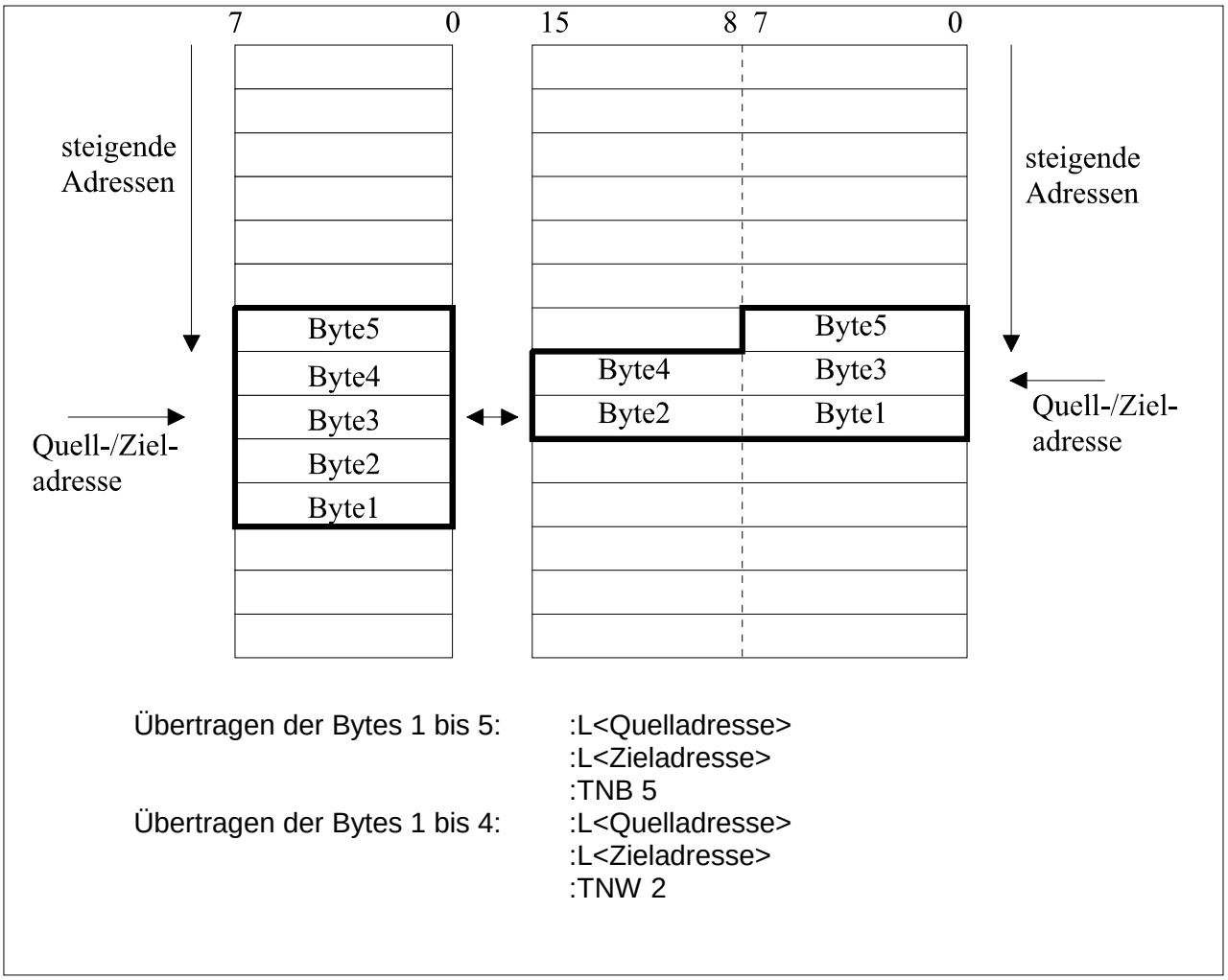
Besonderheiten

Pseudobefehlsgrenzen bei TNB und TNW

Die Operationen TNB und TNW sind langlaufende Operationen, die sogenannte "Pseudobefehlsgrenzen" enthalten: D.h.: Die Datenübertragung erfolgt je nach Quell- und Zielbereich in Teilblöcken unterschiedlicher Größe. Wenn nun während der Übertragung eines Teilblocks ein Fehler (z.B. Zyklusfehler) oder eine Unterbrechung (z.B. durch Weck- oder Prozessalarm) auftritt, so wird am Ende dieses Teilblocks an der Pseudobefehlsgrenze der entsprechende Organisationsbaustein eingeschachtelt. Voraussetzung für den Aufruf des Prozessalarm-OBs oder eines Weckalarm-OBs an einer Pseudobefehlsgrenze ist, dass im DX 0 "Unterbrechbarkeit an Befehlsgrenzen" eingestellt ist.

Tritt während der Übertragung einmal oder mehrmals Quittungsverzug und/oder Adressierfehler auf, so werden zuerst alle Teilblöcke übertragen und dann vor der Ausführung der nächsten Operation einmalig der dafür vorgesehene Fehler-Organisationsbaustein aufgerufen (bei QVZ und ADF gleichzeitig nur der QVZ-OB). Als Fehleradresse wird immer die Adresse angegeben, an der zuerst ein Fehler aufgetreten ist. Da TNB und TNW dekrementierend arbeiten, ist dies bei mehrfachen Fehleradressen die höchste Fehleradresse in dem Bereich, in dem zuerst ein Fehler aufgetreten ist.

TNB und TNW zwischen 8- und 16-bit-Speicherbereichen



Übertragen der Bytes 1 bis 5:
:L<Quelladresse>
:L<Zieladresse>
:TNB 5

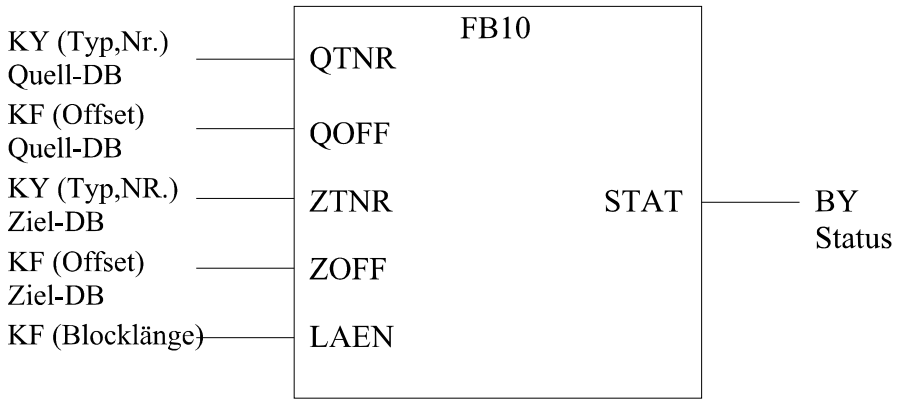
Übertragen der Bytes 1 bis 4:
:L<Quelladresse>
:L<Zieladresse>
:TNW 2

Beispiel für Übertragen von Speicherblöcken

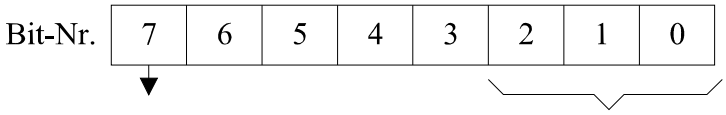
Aufgabenstellung

Kopieren eines Blocks von max. 4095 Datenworten aus einem DB- oder DX-Datenbaustein in einen anderen DB- oder DX-Datenbaustein. Der Anfang des Blocks wird innerhalb des Quell- und Ziel-Datenbausteins durch je einen Offset-Wert zwischen 0 und 4095 festgelegt.

Das Programm wird im Baustein FB10 hinterlegt.



Vor dem Kopieren werden die Eingangsparameter geprüft. Im Fehlerfall wird im Ausgangsparameter STAT das Bit Nr. 7 = 1 gesetzt und in den Bits Nr. 0 bis Nr. 2 die Fehlerart angegeben:



0=kein Fehler
1=Fehler

Fehlerart:
1=Quell-DB=Ziel-DB
2=Offset oder Länge > 4095
3=Quell-DB nicht vorhanden oder unzulässig
4=Quell-DB zu kurz
5=Ziel-DB nicht vorhanden oder unzulässig
6=Ziel-DB im Nur-Lese-Speicher (EPROM-Modul)
7=Ziel-DB zu kurz

Programmstruktur

Der FB10 gliedert sich in fünf Programmabschnitte mit folgenden Aufgaben:

– Eingangsparameter:

a) Prüfen, ob Quell- und Ziel-Datenbaustein nicht denselben Typ und dieselbe Nummer haben.

b) Prüfen, ob die Eingangsparameter "Quell-Offset", "Ziel-Offset" und "Blocklänge" kleiner 4096 sind.

– Quell-Datenbaustein:

a) Prüfen, ob der Quell-Datenbaustein vorhanden und lang genug ist.

b) Berechnen der absoluten Adresse des letzten Datenworts im Ziel-Block.

– Ziel-Datenbaustein:

a) Prüfen, ob der Ziel-Datenbaustein vorhanden und lang genug ist und ob er im Schreib-Lese-Speicher (RAM-Modul oder DB-RAM) liegt.

b) Berechnen der absoluten Adresse des letzten Datenworts im Ziel-Block.

– Transfer:

Kopiervorgang durchführen mit Hilfe des TNW-Befehls.

Blöcke mit mehr als 255 Worte werden in Teilblöcken zu je 128 Worte übertragen (Befehl TNW 128).

– Anzeige:

Versorgen des Ausgangsparameters "Status" entsprechend dem Ergebnis der vorgenommenen Prüfungen.

Belegte Speicherzellen:

MW 242	Endadresse des Datenziels
MW 244	Endadresse der Datenquelle
MW 246	Blocklänge
MW 248	Offset im Ziel-Datenbaustein
MW 250	Typ und Nummer des Ziel-Datenbausteins
MW 252	Offset im Quell-Datenbaustein
MW254	Typ und Nummer des Quell-Datenbausteins
BS 60	Teil-Block-Zähler

Programmierung des FB10:

Hinweis: Soll ab Datenwort DW 0 kopiert werden, entfallen die fett-kursiv gedruckten Programmteile. Es wird kein Offset-Wert angegeben.

```
  FB10
NETZWERK 1
NAME      :DB-DB-TR                          DATENBAUST:-DATENBAUST:-
                                             TRANSFER
BEZ       :QTNR          E/A/D/B/T/Z:  D     KM/KH/KY/KC/KF/KT/KZ/KG: KY
BEZ       :QOFF          E/A/D/B/T/Z:  D     KM/KH/KY/KC/KF/KT/KZ/KG: KF
BEZ       :ZTNR          E/A/D/B/T/Z:  D     KM/KH/KY/KC/KF/KT/KZ/KG: KY
BEZ       :ZOFF          E/A/D/B/T/Z:  D     KM/KH/KY/KC/KF/KT/KZ/KG: KF
BEZ       :LAEN          E/A/D/B/T/Z:  D     KM/KH/KY/KC/KF/KT/KZ/KG: KF
BEZ       :STAT          E/A/D/B/T/Z:  A     BI/BY/W/D: BY
          :
          :                    BEGINN EINGANGS-PARAMETER
          :LW       =QTNR      TYP (DB/DX) UND NUMMER DES
          :T        MW 254     QUELL-DATENBAUSTEINS
          :LW       =ZTNR      TYP (DB/DX) UND NUMMER DES
          :T        MW 250     ZIEL-DATENBAUSTEINS
          :!=F                 QUELL-DB = ZIEL-DB?
          :SPB      =F001      SPRUNG, FALLS JA
          :
          :
          :
          :LW       = QOFF     OFFSET IM QUELL-
          :T        MW 252     DATENBAUSTEIN
          :LW       = ZOFF     OFFSET IM ZIEL-
          :T        MW 248     DATENBAUSTEIN
          :OW
          :LW       =LAEN      LAENGE (ANZAHL DATENWOERTER) DES
          :T        MW 246     ZU TRANSFERIERENDEN BLOCKES
          :                    (BLOCK-LAENGE)
          :OW                  ODER (QUELL-OFFSET/ZIEL-OFFSET)
          :L        KH F000    LAENGE >>= 4096?
          :UW                  SPRUNG, FALLS JA
          :SPP      =F002      END EINGANGS-PARAMETER
          :
          :
          :
          :
          :
                               BEGINN QUELL-DATENBAUSTEIN
          :L        MW 254     TYP UND NUMMER DES QUELL-DATENBAUSTEINS
          :SPA      OB181      DATENBAUSTEIN TESTEN
          :SPB      =F003      SPRUNG, FALLS BST.-TEST NEGATIV
          :TAK                 A1: ANZAHL DW, A2: ADRESSE
          :ENT                 A3: ADRESSE
          :L        MW 252     OFFSET IM QUELL-DATENBAUSTEIN
          :ENT                 A3: ANZAHL DW, A4: ADRESSE
          :L        MW 246     BLOCK-LAENGE
          :+F                  OFFSET + BLOCK-LAENGE
          :<<F                 ANZ. DW <OFFSET + BL.-LAENGE?
          :SPB      =F004      SPRUNG, FALLS JA
          :L        KB 1       A2: OFFSET + BL.-L., A3: ADRESSE
          :-F                  OFFSET + BLOCK-LAENGE - 1
          :+F                  OFFSET + BL.-L. -1 + ADRESSE
          :T        MW 244     END-ADRESSE DER DATEN-QUELLE
          :                    END QUELL-DATENBAUSTEIN
          :
          :
```

Fortsetzung ...

... Fortsetzung

```
                :
                :
                :                       BEGINN ZIEL-DATENBAUSTEIN
                :L      MW 250          TYP U. NUMMER DES ZIEL-DATENBAUSTEINS
                :SPA    OB181           DATENBAUSTEIN TESTEN
                :SPB    =F005           SPRUNG, FALLS BST.-TEST NEGATIV
                :SPM    =F006           SPRUNG, FALLS BST. IM EPROM
                :TAK                    A1: ANZAHL DW, A2: ADRESSE
                :ENT                    A3: ADRESSE
                :L      MW248           OFFSET IM ZIEL- DATENBAUSTEIN
                :ENT                    A3: ANZAHL DW, A4: ADRESSE
                :L      MW 246          BLOCK-LAENGE
                :+F                     OFFSET + BLOCK-LAENGE
                :<<F                    ANZ. DW <OFFSET + BL.-LAENGE ?
                :SPB    =F007           SPRUNG, FALLS JA
                :L      KB 1            A2: OFFSET + BL.-L., A3: ADRESSE
                :-F                     OFFSET + BLOCK-LAENGE - 1
                :+F                     OFFSET + BL.-L. - 1 + ADRESSE
                :T      MW 242          END-ADRESSE DES DATEN-ZIELS
                .                       END ZIEL-DATENBAUSTEIN
                .
                .
                :                       BEGINN TRANSFER
                :L      KB 0            VERGLEICHSWERT
                :L      MB 246          BLOCK-LAENGE, HIGH-BYTE
                :!=F                    BLOCK-LAENGE>>= 256 WORTE ?
                :SLW    1               MULTIPL. MIT 2, ANZAHL TEIL-
                :T      BS 60           BLOECKE MIT JE 128 WORTEN
                :L      MW 244          END-ADRESSE DER DATEN-QUELLE
                :L      MW 242          END-ADRESSE DES DATENZIELS
                :SPB    =REST           SPRUNG, FALLS BLOCK-L. <<256 W.
SCHL    :TNW    128                     TRANSFER EINES TEIL-BLOCKS
                :ADD    KF -128         QUELL-END-ADRESSE UM LAENGE DES
                :TAK                    TEIL-BLOCKS REDUZIEREN
                :ADD    KF -128         ZIEL-END-ADRESSE UM LAENGE DES
                :TAK                    TEIL-BLOCKS REDUZIEREN
                :SPA    OB160           ZAEHL-SCHLEIFE
                :SPB    =SCHL           SPRUNG, FALLS NICHT ALLE
                :                       TEIL-BLOECKE TRANSFERIERT
REST    :B      MW 246                  BLOCK-LAENGE, LOW-BYTE
                :TNW    0               REST-BLOCK TRANSFERIEREN
                :                       END TRANSFER
                :
                :
                :                       BEGINN ANZEIGE
                :L      KB 0            KENNUNG 00 (HEX.): KEIN FEHLER
ENDE    :T      =STAT                   AUSGANGS-PARAMETER STATUS/FEHLER
                :BEA
F001    :L      KB 129                  FEHLER-KENNUNG 81 (HEX.):
                :SPA    =ENDE           QUELL-DB = ZIEL-DB
F002    :L      KB 130                  FEHLER-KENNUNG 82 (HEX.):
                :SPA    =ENDE           OFFSET ODER <N> LAENGE >>= 4096
F003    :L      KB 131                  FEHLER-KENNUNG 83 (HEX.):
                :SPA    =ENDE           QUELL-DB UNZULAESSIG
F004    :L      KB 132                  FEHLER-KENNUNG 84 (HEX.):
                :SPA    =ENDE           QUELL-DB ZU KURZ
F005    :L      KB 133                  FEHLER-KENNUNG 85 (HEX.):
                :SPA    =ENDE           ZIEL-DB UNZULAESSIG
F006    :L      KB 134                  FEHLER-KENNUNG 86 (HEX.):
                :SPA    =ENDE           ZIEL-DB IM NUR-LESE-SPEICHER
F007    :L      KB 135                  FEHLER-KENNUNG 87 (HEX.):
                :SPA    =ENDE           ZIEL-DB ZU KURZ
                :                       END ANZEIGE
                :BE
```

Operationen mit dem Basisadressregister (BR-Register)

Hinweis!

Mit den Befehlen LIR und TIR ist bei der CPU 928C kein Zugriff auf den Anwenderspeicher möglich! Verwenden Sie hier die Befehle LDI und TDI.

Anwendung

Das Basisadressregister (32 bit) ermöglicht Ihnen Adressrechnungen und indirekte Register-Lade- und Transferoperationen ohne Verwendung der AKKUs für die Adressierung.

Zugegriffen wird auf die Speicherzelle, deren absolute Adresse sich als Summe von BR-Registerinhalt und einer Konstanten errechnet:

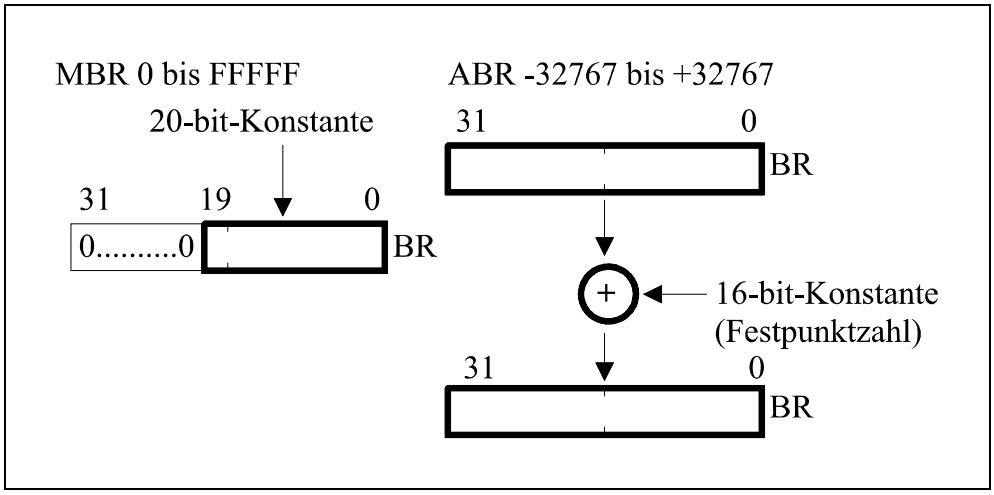
Absolute Adresse = BR-Registerinhalt + Konstante

Operationen

Operation	Operand	Funktion
MBR	Konstante (0h bis FFFFh)	das BR-Register mit einer 20-bit-Konstanten laden [1)]
ABR	Konstante (-32768 bis +32767)	eine 16-bit-Konstante zum Inhalt des BR-Registers addieren

[1)] Die Bits 2^{20} bis 2^{31} des BR-Registers werden gleich '0' gesetzt

Laden des BR-Registers



Veränderung des BR-Registers

- Das BR-Register bleibt erhalten, wenn durch eine Sprunganweisung ("SPA FB"/"SPB FB") in einem anderen Baustein derselben Programmbearbeitungsebene fortgesetzt wird.
- Das BR-Register bleibt nach Einschachtelung einer anderen Programmbearbeitungsebene erhalten.
- Bei Aufruf einer anderen Programmbearbeitungsebene durch das Systemprogramm wird das BR-Register auf '0' gesetzt.

Transfer-operationen zwischen Registern

Anwendung

Die in diesem Abschnitt aufgeführten Operationen können Sie dazu benutzen, schnell Werte zwischen den Registern AKKU 1, STEP-Adresszähler (SAZ) und Basisregister (BR) auszutauschen.

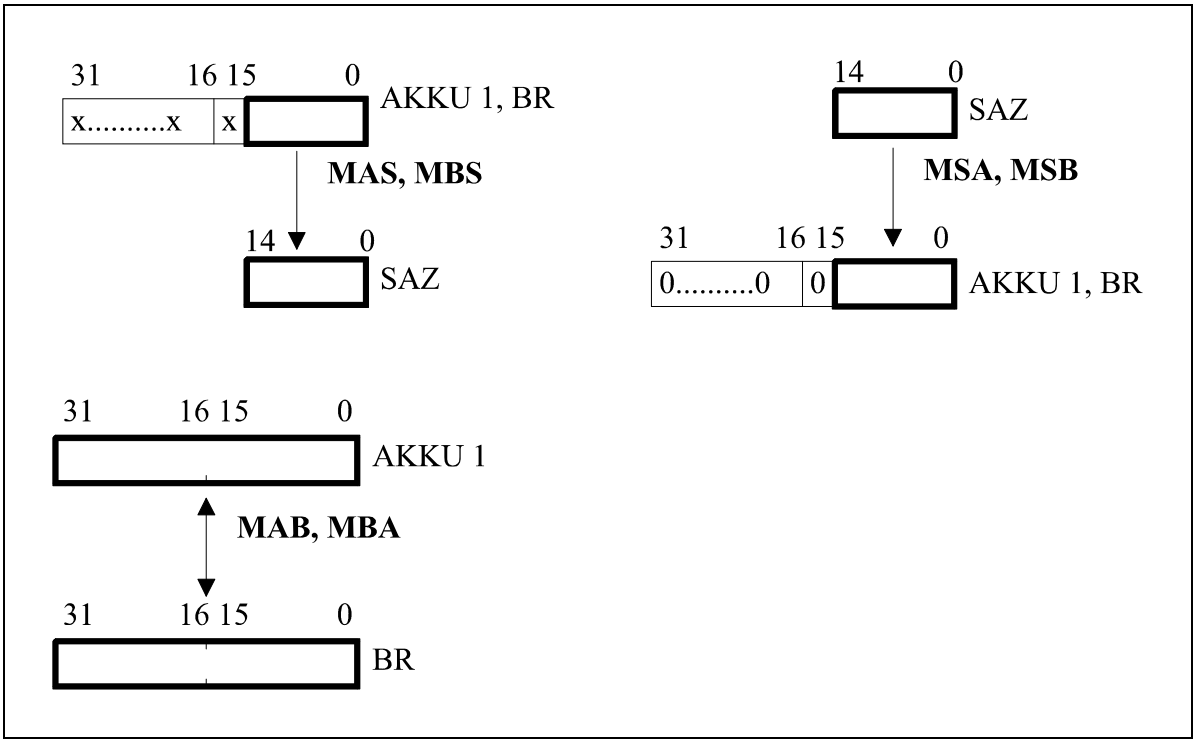
Operationen

Operation	Operand	Funktion
MAS	-	den Inhalt des AKKU 1 (Bit 2^0 bis 2^{14}) in den STEP-Adresszähler transferieren
MAB	-	den Inhalt des AKKU 1 (Bit 2^0 bis 2^{31}) in das Basisadressregister transferieren
MSA	-	den Inhalt des STEP-Adresszählers in den AKKU 1 transferieren [1)]
MSB	-	den Inhalt des STEP-Adresszählers in das Basisadressregister transferieren [1)]
MBA	-	den Inhalt des Basisadressregisters in den AKKU 1 transferieren
MBS	-	den Inhalt des Basisadressregisters (Bit 2^0 bis 2^{14}) in den STEP-Adresszähler transferieren

[1)] Die Bits 2^{15} bis 2^{31} werden auf '0'gesetzt.

Wie die Register bei den Operationen verändert werden, zeigt Ihnen das folgende Bild:

Register-Register-Transferoperationen



Zugriffe auf den lokalen Speicher

Anwendung

Die folgenden Operationen ermöglichen den Zugriff auf den lokalen, wortweise organisierten Speicher über eine absolute Speicheradresse. Die Absolutadresse ist die Summe vom BR-Registerinhalt und der im Befehl enthaltenen 16-bit-Konstanten (-32768 bis +32767).

Operationen

Operation	Operand	Beschreibung
LRW	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt [1] des BR-Registers addieren und das so adressierte Wort in den AKKU-1-L laden
LRD	Konstante (-32768 bis +32767)	die angegeben Konstante zum Inhalt [1] des BR-Registers addieren und das so adressierte Doppelwort in den AKKU 1 laden
TRW	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort transferieren
TRD	Konstante (-32768 bis +32767)	die angegebene Konstante zum Inhalt des BR-Registers addieren und en Inhalt des AKKU 1 in das so adressierte Doppelwort transferieren

[1] AKKU 2_{neu} = AKKU 1_{alt}

Zulässiger Adressbereich

Die Absolutadresse muss liegen

- bei LRW, TRW:
  zwischen 0000h und E3FFh oder E800h und EDFFh,
- bei LRD, TRD:
  zwischen 0000h und E3FEh oder E800h und EDFEh.

Fehlerreaktion

Liegt die errechnete Adresse der Speicherzelle nicht im zulässigen Adressbereich, so erkennt die CPU einen Laufzeitfehler und ruft den OB31 auf, falls er geladen ist. Ist der OB31 nicht geladen, so geht die CPU in den Stoppzustand.

In beiden Fällen sind im AKKU-1-L Fehlerkennungen hinterlegt, die den aufgetretenen Fehler näher erläutern.

Zugriffe auf den Kachelspeicher

Anwendung

Mit den folgenden Operationen können Sie auf byte- oder wortweise organisierte Kacheln über eine absolute Speicheradresse zugreifen. Die Absolutadresse ist die Summe vom BR-Register-Inhalt und der im Befehl enthaltenen Konstanten (-32768 bis +32767).

Prinzipieller Ablauf des Kachelzugriffs

Der Globalbereich enthält im Bereich der Adressen F400h bis FBFFh ein "Fenster" zum Einblenden eines von max. 256 Speicherbereichen (= Kacheln). Eine Kachel belegt max. 2K Adressen und kann byteweise oder wortweise organisiert sein. Vor jedem Zugriff auf den Kachelbereich wird eine der 256 Kacheln durch Eintrag ihrer Kachelnummer in das Select-Register (Kachel-Adressregister) ausgewählt. Der Vorgang "Beschreiben des Select-Registers und anschließender Zugriff auf den Kachelbereich" ist ununterbrechbar.

Vor jedem Zugriff (Laden/Transferieren) auf den Kachelbereich muss eine der 256 Kacheln aufgeschlagen werden. Dazu übergeben Sie die Nummer der aufzuschlagenden Kachel im AKKU-1-L; diese Nummer wird durch den Befehl ACR in das CPU-interne Kachel-Register eingetragen. Alle folgenden Kacheloperationen schreiben vor dem Kachelzugriff den Inhalt des Kachel-Registers in das Select-Register der entsprechenden Baugruppen auf dem Rückwand-Bus.

Veränderung des Kachel-Registers

- Das Kachel-Register bleibt erhalten, wenn ein anderer Baustein derselben Programmbearbeitungsebene aufgerufen wird.
- Wenn das Kachelregister in einem Baustein verändert wird, so bleibt sein Wert erhalten, wenn am Bausteinende in den aufrufenden Baustein zurückgesprungen wird.
- Das Kachelregister wird nach Einschachtelung einer anderen Programmbearbeitungsebene durch das Systemprogramm mit demselben Wert geladen, den es vor der Einschachtelung hatte.
- Bei Aufruf einer anderen Programmbearbeitungsebene durch das Systemprogramm wird das Kachelregister '0' gesetzt.

Aufschlagen einer Kachel

Operation	Parameter	Beschreibung
ACR		diejenige Kachel aufschlagen, deren Nummer im AKKU-1-L steht, zulässige Werte: 0 bis 255

Fehlerreaktion

Die Kachelnummer muss zwischen 0 und 255 liegen. Ist dies nicht der Fall, so erkennt die CPU einen Laufzeitfehler und ruft den OB31 auf, falls er geladen ist. Ist der OB31 nicht geladen, so geht die CPU in den Stoppzustand. In beiden Fällen sind im AKKU-1-L Fehlerkennungen hinterlegt, die den aufgetretenen Fehler näher erläutern.

Lade- und Transferoperationen für byteweise organisierte Kacheln

Operation	Operand	Beschreibung
LB CB	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Byte in der aufgeschlagenen Kachel in den AKKU-1-LL laden [1)3)]
LB CW	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in der aufgeschlagenen Kachel in den AKKU-1-L laden [2)3)]
LB CD	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in der aufgeschlagenen Kachel in den AKKU 1 laden [3)]
TB CB	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-LL in das so adressierte Byte in der aufgeschlagenen Kachel transferieren
TB CW	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort in der aufgeschlagenen Kachel transferieren
TB CD	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort in der aufgeschlagenen Kachel transferieren

[1)] AKKU-1-LH und AKKU-1-H werden auf '0' gesetzt.

[2)] AKKU-1-H wird auf '0' gesetzt.

[3)] AKKU 2_{neu} :=AKKU 1_{alt}

Zulässiger Adressbereich

Die Absolutadresse muss liegen

- bei LB CB, TB CB: zwischen F400H und FBFFH,
- bei LB CW, TB CW: zwischen F400H und FBFEH,
- bei LB CD, TB CD: zwischen F400H und FBFCH.

Fehlerreaktion

Liegt die errechnete Byteadresse nicht im zulässigen Adressbereich, so erkennt die CPU einen Laufzeitfehler und ruft den OB31 auf, falls er geladen ist. Ist der OB31 nicht geladen, so geht die CPU in den Stoppzustand. In beiden Fällen sind im AKKU-1-L Fehlerkennungen hinterlegt, die den aufgetretenen Fehler näher erläutern.

Lade- und Transferoperationen für wortweise organisierte Kacheln

Operation	Operand	Beschreibung
LW CW	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Wort in der aufgeschlagenen Kachel in den AKKU-1-L laden [1)]
LW CD	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und das so adressierte Doppelwort in der aufgeschlagenen Kachel in den AKKU 1 laden [2)]
TW CW	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU-1-L in das so adressierte Wort in der aufgeschlagenen Kachel transferieren
TW CD	-32768 bis 32767	die angegebene Konstante zum Inhalt des BR-Registers addieren und den Inhalt des AKKU 1 in das so adressierte Doppelwort in der aufgeschlagenen Kachel transferieren

[1)] AKKU-1-H wird auf '0' gesetzt.

[2)] AKKU 2_{neu} := AKKU 1_{alt}

Zulässiger Adressbereich

Die Absolutadresse muss liegen

- bei LW CW, TW CW: zwischen F400h und FBFFh,
- bei LW CD, TW CD: zwischen F400h und FBFEh.

Fehlerreaktion

Liegt die errechnete Adresse der Speicherzelle nicht im zulässigen Adressbereich, so erkennt die CPU einen Laufzeitfehler und ruft den OB31 auf, falls er geladen ist. Ist der OB31 nicht geladen, so geht die CPU in den Stoppzustand.

In beiden Fällen sind im AKKU-1-L Fehlerkennungen hinterlegt, die den aufgetretenen Fehler näher erläutern.

Teil 10 Mehrprozessorbetrieb und -kommunikation

Mehrprozessorbetrieb wird zur Zeit noch nicht unterstützt.

Teil 11 PG-Schnittstellen und -Funktionen

Überblick

Dieses Kapitel informiert Sie darüber, wie Sie an die CPU 928 ein PG koppeln können und welche Hilfen Ihnen die PG-Software für den Test Ihres Anwenderprogramms bietet.

Falls Sie nur die Standard-PG-Schnittstelle (SI 1) benutzen, brauchen Sie den Abschnitt "Serielle Kopplung" nicht zu lesen.

Diese Abschnitte informieren Sie darüber, wie Sie über weitere Schnittstellen ein PG mit Ihrer CPU 928 koppeln können.

Sie erfahren darin ferner, was Sie bei der Benutzung der PG-Funktionen auf beiden Schnittstellen beachten müssen.

Die Beschreibungen in diesem Kapitel beziehen sich auf die Online-Funktionen des MC5-Pakets von VIPA und des STEP®5-Progammier-Pakets von Siemens.

Nachfolgend sind beschrieben:

- PG-Funktionalität
- Hilfen für Inbetriebnahme und Test
- Auskunft- und Statusfunktionen
- Tätigkeiten an Kontrollpunkten
- Serielle Kopplung und Parallelbetrieb von 2 seriellen Schnittstellen

Inhalt

Thema **Seite**

Teil 11 PG-Schnittstellen und -Funktionen 11-1

Übersicht 11-2

PG-Funktionen 11-3

Tätigkeiten an Kontrollpunkten 11-16

Serielle Kopplung.....11-17

Parallelbetrieb von zwei seriellen PG-Schnittstellen.....11-17

Übersicht

PG-Kopplung

Das Laden und Testen Ihres Anwenderprogramms führen Sie mit Hilfe der Online-Funktionen Ihres Software-Pakets durch.

Dazu müssen Sie Ihre CPU mit dem PG koppeln. Für diese Kopplung stehen Ihnen folgende Schnittstellen zur Verfügung:

- Kopplung über die serielle Standard-Schnittstelle "PG-AG",
- Kopplung über die 2. Serielle Schnittstelle der CPU 928.

Die PG-Funktionen laufen auf beiden seriellen Schnittstellen parallel ab.

PG-Funktionen

Die PG-Funktionen bieten Ihnen folgende Unterstützung für die Inbetriebnahme und den Test Ihres Programms:

Übersicht

Auskunft	
"Speicherausbau"	höchste nutzbare Adresse und letzte belegte Adresse des Anwenderspeichers
"Bausteinliste"	Liste der geladenen Bausteine
"Ausgabe Adresse"	Inhalt von Speicherwörtern/bytes und Peripheriebytes anzeigen
Speicherverwaltung	
"Urlöschen"	Speicher komplett löschen
"Speicher komprimieren"	freien Speicherplatz zusammenfassen
"Baustein übertragen/löschen"	Bausteine verwalten
Programmtest	
"Start/Stopp"	CPU starten/anhalten
"Status"	Befehlsfolge in einem Baustein testen
"Bearbeitungskontrolle"	einzelne Programmschritte testen
"Status Variablen"	Signalzustand von Prozessvariablen anzeigen
"Steuern"	Ausgangssignale im Stopp ausgeben
"Steuern Variablen"	Prozessvariable anzeigen/verändern

PG-Funktionen

Hinweis,

Die in diesem Abschnitt für die PG-Funktionen verwendeten Begriffe können u.U. von den Begriffen in Ihrer PG-Software abweichen. Sie beziehen sich auf die Online-Funktionen des MC5-Pakets von VIPA und des STEP®5-Progammier-Pakets von Siemens.

Aufrufen und Bedienen

Wie Sie die einzelnen PG-Funktionen aufrufen und bedienen, lesen Sie bitte im Handbuch zu Ihrem Programmiergerät nach.

Achtung!

Während der Speicher mittels des OB 186 komprimiert wird, wird die PG-Funktion "Speicher komprimieren" abgewiesen. Andere PG-Funktionen sind nur eingeschränkt zulässig.

Während PG-Funktionen aktiv sind, wird der OB 186 abgewiesen

Ausführung

Im Automatisierungsgerät werden die PG-Funktionen an definierten Punkten ausgeführt. Hierbei gibt es Punkte im Systemprogramm (= Systemkontrollpunkte) und Punkte im Anwenderprogramm (= Anwenderkontrollpunkte).

Systemkontrollpunkte

Im Betriebszustand STOP existiert der Systemkontrollpunkt "Stop", der regelmäßig aufgerufen wird.

Im Betriebszustand RUN wird der Systemkontrollpunkt "Zyklus" am Ende der Programmbearbeitungsebene ZYKLUS vor der Prozessabbildaktualisierung aufgerufen.

Befindet sich die CPU im WARTEZUSTAND, so wird dort regelmäßig der Systemkontrollpunkt "Wartezustand" aufgerufen.

Zusätzlich gibt es einen zeitbedingten Systemkontrollpunkt "Asynchron". Dieser Systemkontrollpunkt wird asynchron während der Programmbearbeitung eingeschachtelt.

Anwenderkontrollpunkte

Bei den Testfunktionen "Status Baustein" und "Bearbeitungskontrolle" werden Anwenderkontrollpunkte verwendet. Ein Anwenderkontrollpunkt wird aufgerufen, wenn ein Befehl ausgeführt ist, der vom PG markiert ist.

Betriebszustand WARTEZUSTAND

Bisher sind Ihnen die Betriebszustände STOP, ANLAUF und RUN bekannt.

Bei der PG-Funktion "Bearbeitungskontrolle" nimmt die CPU einen weiteren Betriebszustand ein: den WARTEZUSTAND.

Wenn die CPU sich im WARTEZUSTAND befindet, können noch weitere PG-Funktionen aufgerufen werden.

Eigenschaften

- Im Wartezustand wird das Anwenderprogramm nicht bearbeitet.
- LEDs auf der Frontplatte:

RUN-LED:	aus
STOP-LED:	aus
BASP-LED:	an

- Alle Zeitzellen sind "eingefroren", d.h. es laufen keine Timer (die Zeitzellen werden nicht verändert). Ebenso bleiben alle Systemzeiten stehen, wie die der Regelung und die der zeitgesteuerten Bearbeitung. Nach Verlassen des Wartezustands laufen die Timer weiter.
- Unterbrechungsursachen wie z.B. PEU, BAU oder Stoppschalter werden im Wartezustand registriert. Es erfolgt keine Reaktion darauf.

Unterbrechungen

Wenn im Wartezustand Unterbrechungsursachen registriert worden sind, so werden die dazugehörigen Programmbearbeitungsebenen unmittelbar nach Verlassen des Wartezustands aufgerufen.

Tritt NAU auf, so wird der Wartezustand verlassen und die Online-Funktion BEARBEITUNGSKONTROLLE abgebrochen. Nach NETZ EIN ist in den Steuerbits BARBEND angekreuzt. Der Stoppzustand kann nur mit NEUSTART verlassen werden.

Auskunft

Speicherausbau

Die PG-Funktion "Speicherausbau" zeigt Ihnen am PG die höchste nutzbare Adresse des Anwenderspeichers an (bei EPROM-Betrieb wird '0' angezeigt) und die letzte mit Bausteinen des Anwenderprogramms belegte Adresse des Speichermoduls.

Ausgabe Adresse

Mit der Funktion "Ausgabe Adresse" können Sie sich am PG den Inhalt von Speicher- und Peripherieadressen hexadezimal anzeigen lassen. Sie können alle Adressen ansprechen. Im Bereich des Prozessabbildes wird kein ADF ausgelöst, wenn zum entsprechenden Byte keine Peripheriebaugruppe vorhanden ist; im Peripheriebereich entsteht kein QVZ.

Im byteadressierbaren Bereich (Merker, Prozessabbild) wird das High-Byte als 'FF' angezeigt.

Im Peripheriebereich wird bei quittierenden Adressen das Highbyte als '00' ausgegeben. Quittiert eine Peripherieadresse nicht, wird das Highbyte als 'FF' angezeigt.

Speicher- und Übertragungs-funktionen

Urlöschen

Sie können mit der Funktion "Löschen alle Bausteine" eine CPU vom PG aus urlöschen. Dabei wird das Urlöschen unbedingt durchgeführt (siehe die Teile "Betriebszustände" und "Programmbearbeitung").

Befindet sich die CPU beim Aufruf von "Löschen aller Bausteine" im Zustand ANLAUF oder RUN, wird zunächst ein Übergang in den Stoppzustand durchgeführt. Dabei wird - wenn geladen - der Organisationsbaustein OB 28 aufgerufen.

Hinweis!

Urlöschen" ist nicht zulässig, solange "Bearbeitungskontrolle" aktiv ist!

Speicher komprimieren

Diese Funktion optimiert die Speicherbelegung durch Bausteine: Der Platz, den ungültig markierte Bausteine belegen, wird so durch die vorhandenen gültigen Bausteine überschrieben (der Baustein wird auf einen anderen Speicherplatz umgeladen), dass diese vom Anfang des Speichers an hintereinanderliegende Adressen sind.

Dies wird getrennt im Anwenderspeicher und im DB-RAM durchgeführt und kann an den Systemkontrollpunkten "Zyklus" und "Stop" erfolgen.

Bei der CPU 928 ist die Funktion "Speicher komprimieren" im STOP immer möglich, auch wenn der BSTACK nicht leer ist.

Achtung!

Nach "Speicher komprimieren" im STOP ist als Anlaufart nur NEUSTART zulässig. USTACK und BSTACK werden nicht nachgeführt.

Netzspannungs-ausfall beim Komprimieren

Fällt während des Komprimierens die Netzspannung aus, so wird nach Netzwiederkehr der Vorgang abgeschlossen und kein weiterer Baustein umgeladen. Bei erneutem Aufruf von "Speicher komprimieren" wird der Optimierungsvorgang fortgesetzt.

Fehler im Bausteinspeicher

Die Funktion "Speicher komprimieren" erkennt folgende Fehler im Bausteinspeicher:

- falsche Bausteinlänge,
- verfälschtes Muster '7070' im Bausteinkopf,
- ungültiger Bausteintyp (bei OBs ungültige Bausteinnummer).

Die Funktion wird daraufhin abgebrochen und am PG eine Meldung ausgegeben. Sie müssen dann zuerst "Urlöschen"; daraufhin ist der Speicher leer.

Hinweis!

"Speicher komprimieren" ist nicht zulässig, solange "Bearbeitungskontrolle" aktiv ist.

Baustein übertragen

Mit dieser Funktion übertragen Sie neue oder schon vorhandene Code- und Datenbausteine in den Anwenderspeicher bzw. in das interne DB-RAM der CPU.

Ist ein Baustein bereits im Anwenderspeicher bzw. im internen DB-RAM der CPU vorhanden, so wird er als ungültig markiert und der neue erhält die Gültigmarkierung. Ein Baustein wird nur dann für ungültig erklärt, wenn er momentan nicht bearbeitet wird.

Baustein löschen

Mit dieser Funktion erklären Sie Code- und Datenbausteine im Anwenderspeicher der CPU für ungültig. Ein Baustein wird nur dann für ungültig erklärt, wenn er momentan nicht bearbeitet wird.

Den Platz, an den Sie diese Bausteine gelegen haben, können Sie über die Funktion "Speicher komprimieren" für andere Bausteine ausnutzen.

Programmtest

Start/Stopp

Die PG-Bedienung entspricht der manuellen Bedienung.

Mit dem Aufruf der PG-Funktion "Stopp" im Betriebszustand RUN bringen Sie das Automatisierungsgerät in den Stoppzustand.

Bei der CPU, an der das PG angeschlossen ist, sehen Sie folgendes Bild:

STOP-LED: an

BASP-LED: an

In den Steuerbits ist PG-STP angekreuzt.

Sie können eine CPU mit NEUSTART oder WIEDERANLAUF starten. Die CPU verlässt den Stoppzustand.

Mit der PG-Funktion "Start" können Sie den gewünschten Anlauf durchführen und die CPU starten.

- PG-Funktion NEUSTART:
  Es wird ein MANUELLER NEUSTART der CPU durchgeführt.

- PG-Funktion WIEDERANLAUF:

Es wird immer ein MANUELLER WIEDERANLAUF durchgeführt.

Status Baustein

Mit Hilfe der Funktion "Status" testen Sie an beliebiger Stelle im Anwenderprogramm zusammenhängende Befehlsfolgen in einem Baustein.

Zu jedem ausgeführten Befehl im Baustein werden die aktuellen Signalzustände der Operanden, die AKKU-Inhalte, das VKE usw. am Programmiergerät ausgegeben (Schrittbetrieb). Auch die Parametrierung von Funktionsbausteinen kann auf diese Weise getestet werden:

Angezeigt werden die Signalzustände der Aktualoperanden.

Funktion aufrufen und Haltepunkt vorgeben

Wenn Sie die Funktion "Status" am PG aufrufen und Bausteinart und Bausteinnummer (evtl. mit Schachtelreihenfolge und Suchbegriff) des zu testenden Bausteins eingeben, so geben Sie damit einen sog. "Haltepunkt" vor.

Bei Aufruf der Funktion während der Programmbearbeitung im RUN wird die Programmbearbeitung solange fortgesetzt, bis der durch den vorgegebenen Haltepunkt markierte Befehl in der richtigen Schachtelfolge erreicht ist. Danach werden die überwachten Befehle jeweils bis zur Befehlsgrenze ausgeführt und die Ergebnisse der Befehlsbearbeitung am PG ausgegeben.

Aufruf im STOP

Die Funktion "Status" lässt sich auch im STOP aufrufen. Danach ist sowohl ein NEUSTART als auch ein MANUELLER WIEDERANLAUF möglich. Die CPU bearbeitet daraufhin das Anwenderprogramm bis zum vorgegebenen Haltepunkt. Dann werden die Daten zu der gewünschten Befehlsfolge ausgegeben. Somit eignet sich die Funktion "Status" zum Beispiel auch dazu, das Anwenderprogramm im Anlauf oder im ersten Zyklus zu testen.

Hinweis!

Die Ergebnisse der Befehlsbearbeitung werden nicht in jedem Programmzyklus ausgegeben.

Einschachtelungen und Unterbrechungen

Eine durch einen vorgegebenen Haltepunkt markierte Befehlsfolge wird vollständig durchlaufen, auch wenn zwischendurch an einer Befehlsgrenze eine andere Programmbearbeitungsebene (z.B. ein Fehler-OB, ein Prozess- oder ein Weckalarm) eingeschachtelt und abgearbeitet wird.

Führt in einer eingeschachtelten Programmbearbeitungsebene eine Unterbrechungsursache die CPU in den Stoppzustand, so werden im STOP die Daten bis zu demjenigen Befehl ausgegeben, der als letzter vor der Einschachtelung ausgeführt worden ist. Die Daten der restlichen Befehle werden mit '0' aufgefüllt (auch SAZ = 0).

Wenn die CPU von einem Betriebszustand in den anderen wechselt (z.B. RUN $\rightarrow$ STOP $\rightarrow$ MANUELLER WIEDERANLAUF), bleibt die Funktion weiterhin aufgerufen.

Beendet wird "Status" durch Betätigen der Abbruchtaste am Programmiergerät.

Bearbeitungs-kontrolle

Mit der Funktion "Bearbeitungskontrolle" testen Sie an beliebiger Stelle im Anwenderprogramm einzelne Programmschritte. Dazu halten Sie die Programmbearbeitung an und lassen die CPU dann einen Befehl nach dem anderen bearbeiten. Zu jedem ausgeführten Befehl werden die aktuellen Signalzustände von Operanden, die AKKU-Inhalte, das VKE usw. am Programmiergerät ausgegeben.

Funktion aufrufen und 1. Haltepunkt vorgeben

Zum Aufruf der Funktion "Bearbeitungskontrolle" geben Sie Bausteinart und Bausteinnummer (eventuell mit Schachtelreihenfolge) des zu testenden Bausteins an und markieren am PG den ersten Befehl, dessen Daten ausgegeben werden sollen. Damit geben Sie einen ersten Halte-punkt vor.

In den Steuerbits wird BARB angekreuzt. Die Befehlsausgabe wird gesperrt (BASP-LED = an).

Aufruf im ANLAUF und im RUN

Wenn Sie den 1. Haltepunkt während der **Programmbearbeitung** im ANLAUF oder RUN vorgeben, setzt die CPU die Programmbearbeitung solange fort, bis der durch den vorgegebenen Haltepunkt markierte Befehl erreicht ist. Der Befehl wird noch komplett ausgeführt. (Die Befehle B MW und B DW werden **einschließlich** substituiertem Befehl bearbeitet.)

Anschließend geht die CPU in den WARTEZUSTAND. Dort werden die Daten des markierten und zuletzt bearbeiteten Befehls ausgegeben.

Aufruf im STOP

Auch im STOP können Sie die Funktion BEARBEITUNGSKONTROLLE aufrufen und einen ersten Haltepunkt vorgeben. Die CPU bleibt weiterhin im Stoppzustand. Sie können jetzt sowohl einen NEUSTART als auch einen MANUELLEN WIEDERANLAUF durchführen. Die CPU führt die Programmbearbeitung bis zum markierten Befehl aus und verfährt dann wie oben.

Funktion fortführen und weiteren Haltepunkt vorgeben

Ausgangspunkt:

Die CPU befindet sich im WARTEZUSTAND.

Um die Funktion fortzuführen, haben Sie zwei Möglichkeiten.

1. Sie geben einen Folgehaltepunkt vor:
   Der vorgebenene Haltepunkt wird um einen Befehl verschoben. Die CPU verlässt den WARTEZUSTAND und setzt die Programmbearbeitung um diesen einen Befehl fort. Wenn der Befehl komplett bearbeitet ist, geht die CPU erneut in den WARTEZUSTAND und gibt dort die Daten aus.
   Wird der Folgebefehl jedoch in einer eingeschachtelten Programmbearbeitungsebene erreicht, setzt die CPU die Programmbearbeitung fort. Der Folgepunkt bleibt weiterhin vorgegeben.

Hinweis!

Im Stoppzustand können Sie keinen Folgehaltepunkt vergeben!

2. Sie geben einen neuen Haltepunkt vor:
   Sie geben am PG einen beliebigen anderen Befehl im gleichen oder einem anderen Baustein vor. Die CPU setzt die Programmbearbeitung fort, bis sie den neuen Haltepunkt erreicht. Der Befehl wird vollständig bearbeitet. Dann geht die CPU in den WARTEZUSTAND und gibt dort die Daten aus.
   Sie können die CPU mit Bearbeitungskontrolle auch um einen ganzen Zyklus weiterlaufen lassen (zyklusweise testen). Dafür setzen Sie im WARTEZUSTAND den Haltepunkt auf denselben Befehl wie vorher. Allerdings darf sich der Befehl nicht in einer Programmierschleife befinden. In diesem Fall wird die Schleife einmal durchlaufen; es erfolgt keine Programmbearbeitung über die Zyklusgrenze hinweg.

Hinweis!

Im WARTEZUSTAND können Sie andere Funktionen wie "Ausgabe Buch", "Status Variablen" oder "Steuern Variable" aufrufen. Sobald die Programmbearbeitung nach Verlassen des WARTEZUSTANDs fortgesetzt wird, laufen die Timer und die Systemzeiten weiter, bis wieder ein Haltepunkt erreicht wird.

Haltepunkt zurücknehmen

Ist ein vorgegebener Haltepunkt noch nicht erreicht, so haben Sie die Möglichkeit, diesen nachträglich zurückzunehmen, indem Sie am PG die Abbruchtaste betätigen. Die CPU geht daraufhin in den WARTEZUSTAND. Danach können Sie einen neuen Haltepunkt vorgeben oder "Bearbeitungskontrolle Ende" aufrufen.

Funktion abbrechen

Durch Aufrufen von "Bearbeitungskontrolle Ende" können Sie während der Programmbearbeitung im WARTEZUSTAND oder im STOP die Funktion abbrechen. Die CPU geht in den STOP (bzw. bleibt im STOP). Die STOP-LED blinkt langsam. In den Steuerbits wird BARBEND angekreuzt. Anschließend ist ein NEUSTART erforderlich.

Tritt während der Funktion "Bearbeitungskontrolle" ein Schnittstellenfehler auf (Unterbrechung am PG-Kabel), wird die Funktion abgebrochen wie oben beschrieben.

Einschachtelungen

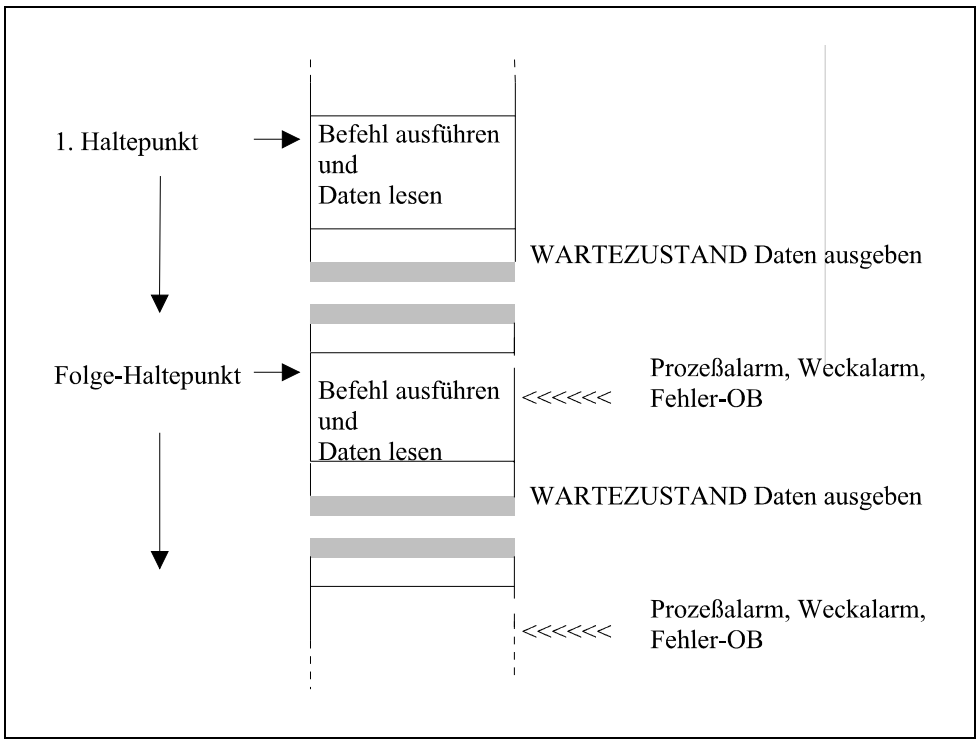
Bei aufgerufener Funktion "Bearbeitungskontrolle" können nach Verlassen des WARTEZUSTANDs andere Programmbearbeitungsebenen eingeschachtelt werden.

Wenn der Befehl am Haltepunkt bearbeitet ist und an dieser Stelle eine andere Programmbearbeitungsebene aufgerufen ist (z.B. ein Fehler-OB oder ein Alarm-OB), so wird diese erst eingeschachtelt und vollständig bearbeitet, wenn der WARTEZUSTAND wieder verlassen wird.

Hinweis!

Die Daten werden an der Befehlsgrenze gelesen und dort ausgegeben. Alle dann evtl. folgenden Einschachtelungen sind noch nicht bearbeitet

Testablauf bei Bearbeitungskontrolle



"

Sind im WARTEZUSTAND Anforderungen wie PEU, Stoppschalter usw. aufgetreten, werden diese nur registriert. Sofort nach Verlassen des WARTEZUSTANDs können diese wirksam werden: Eine Programmbearbeitungsebene wird eingeschachtelt oder eine Unterbrechung führt in den STOP.

Es gilt die Reihenfolge der Ereignisse. Gleichzeitige Anforderungen werden priorisiert.

Hinweis!

Wenn die CPU im WARTEZUSTAND ist und eine Einschachtelung angefordert ist, haben Sie die Möglichkeit, einen Haltepunkt auf einen Befehl der Einschachtelung zu setzen. So können Sie z.B. bei einem Befehl, der einen QVZ auslöst, direkt danach den QVZ-Fehler-OB beobachten.

Unterbrechungen

- Programmbearbeitung (ANLAUF/RUN) → Stoppzustand:
  Tritt während der Programmbearbeitung eine Unterbrechung auf (z.B. Peripherie unklar, STOP-Schalter, Fehler-OB nicht programmiert usw.) und der vorgegebene Haltepunkt ist noch nicht erreicht, so geht die CPU in den Stoppzustand. Wenn danach ein Anlauf (NEUSTART oder MANUELLER WIEDERANLAUF) durchgeführt wird, bleibt die Funktion "Bearbeitungskontrolle" weiterhin aufgerufen; der Haltepunkt ist weiterhin vorgegeben.
- Befehlsbearbeitung am Haltepunkt (ANLAUF/RUN) → Stoppzustand:
  Wenn während der Befehlsbearbeitung am Haltepunkt oder Folgehaltepunkt Stopp-Bedingungen auftreten (STOP-Schalter, Befehl STP, Fehler-OB nicht programmiert), so geht die CPU unmittelbar nach der Befehlsbearbeitung in den STOP.
  Wird im Stoppzustand kein neuer Haltepunkt vorgegeben, geht die CPU nach einem ANLAUF in den WARTEZUSTAND. Die Funktion "Bearbeitungskontrolle" bleibt weiterhin aufgerufen.
- Wartezustand → STOP
  Unterbrechungsursachen, die im WARTEZUSTAND auftreten (z.B. Peripherie unklar, STOP-Schalter) oder vom vorausgegangenen Befehl stammen (Fehler, der in den STOP führt), werden zwar registriert, die CPU bleibt jedoch im WARTEZUSTAND. Erst wenn im WARTEZUSTAND ein neuer Haltepunkt vorgegeben wird und die CPU den WARTEZUSTAND verlässt, bewirken die aufgetretenen Unterbrechungsursachen einen Übergang in den STOP. Der vorgegebene Haltepunkt wird nicht erreicht.
  Wird jetzt ein ANLAUF (NEUSTART oder MANUELLER WIEDERANLAUF) durchgeführt, ist der neue Haltepunkt weiterhin vorgegeben.

Hinweis!

Wird im WARTEZUSTAND der Betriebsartenschalter auf STOP gestellt, so geht die CPU erst nach Verlassen des WARTEZUSTANDs in den STOP.

Führen Unterbrechungsursachen die CPU während der "Bearbeitungskontrolle" in den STOP, bleibt nach einem anschließenden ANLAUF die Funktion "Bearbeitungskontrolle" (und ein evtl. vorgegebener Haltepunkt) weiterhin aktiv!

Status Variablen

Mit der PG-Funktion "Status Variablen" können Sie sich die aktuellen Signalzustände bestimmter Operanden (Prozessvariablen) anzeigen lassen.

Die Funktion aktiviert Systemkontrollpunkte im ZYKLUS, im STOP und im WARTEZUSTAND.

Wenn ein Kontrollpunkt erreicht ist, wird der zu diesem Zeitpunkt aktuelle Signalzustand der gewünschten Prozessvariablen ausgegeben. Sie können alle Prozessvariablen angeben (Eingänge, Ausgänge, Merker, Zeitzellen, Zähler und Datenwörter). Im Bereich des Prozessabbilds wird bei Zugriff auf eine Adresse, zu der keine Peripherie vorhanden ist, kein ADF ausgelöst.

Ablauf während der Programmbearbeitung

Läuft die Funktion im Betriebszustand RUN oder ANLAUF, so wird die Programmbearbeitung fortgesetzt, bis der Systemkontrollpunkt "Zyklus" erreicht ist. Dann werden die Signalzustände der Operanden am Zyklusende abgefragt und angezeigt. Eingänge werden aus dem **Prozessabbild** gelesen. Solange die Funktion nicht abgebrochen wird, werden bei laufender Programmbearbeitung die Signalzustände zyklisch aktualisiert. Die Signalzustände werden dabei nicht an jedem Systemkontrollpunkt abgefragt.

Wird der Systemkontrollpunkt "Zyklus" nicht erreicht, erfolgt keine Ausgabe der Signalzustände (z.B. bei einer Dauerschleife im Anwenderprogramm)!

Ablauf der Funktion im STOP

Wenn die Funktion "Status Variablen" im STOP läuft, werden die Signalzustände der Operanden ausgegeben, wie sie am Systemkontrollpunkt "Stop" vorliegen. Wichtig ist dabei, dass die **Eingänge direkt** von der Peripheriebaugruppe abgefragt und ausgegeben werden. Dadurch lässt sich zum Beispiel testen, ob ein Peripherie-Eingangssignal tatsächlich zur CPU gelangt. Die Ausgänge werden vom Prozessabbild gelesen.

Ablauf der Funktion im WARTEZUSTAND

Die Funktion "Status Variablen" können Sie auch aufrufen, wenn die CPU sich mit der Funktion "Bearbeitungskontrolle" im WARTEZUSTAND befindet. Am Systemkontrollpunkt "Wartezustand" werden die Signalzustände der Operanden abgefragt und ausgegeben. Wie im Stoppzustand werden dabei die Eingänge direkt, die Ausgänge dagegen aus dem Prozessabbild gelesen.

Wechsel des Betriebszustands/ Beenden der Funktion

Wenn die CPU von einem Betriebszustand in den anderen wechselt (z.B. RUN → STOP → MANUELLER WIEDERANLAUF), bleibt die Funktion weiterhin aufgerufen. Beendet wird der "Status Variablen" durch Betätigen der Abbruchtaste am Programmiergerät.

Hinweis!

Die Variablen werden nicht in jedem Programmzyklus ausgegeben.

Steuern

Mit Hilfe der Funktion "Steuern" können Sie die Ausgangsbytes des Automatisierungsgerätes direkt (unter Umgehung des Prozessabbildes) auf einen gewünschten Signalzustand einstellen oder nicht quittierende Signalformer (digitale Peripherie 0 bis 127) erkennen (Meldung am PG). Sie haben die Möglichkeit, die von den Ausgängen versorgten Prozessgeräte ("Aktoren" wie z.B. Motor, Ventil) direkt zu überprüfen und zu steuern.

Hinweis!

Die Funktion "Steuern" ist nur im **STOP** zulässig!

Ablauf der Funktion

Bei Aufruf der Funktion im STOP wird die Befehlsausgabesperre aufgehoben (BASP = inaktiv). Die **gesamte** digitale Peripherie (F000h bis F07Fh) wird gelöscht, wobei jede Adresse mit dem Wert '0' beschrieben wird. Während des Löschens der Peripherie ist die Funktion nicht unterbrechbar.

Die Peripherieausgänge werden byteweise gesteuert, direkt und ohne das Prozessabbild der Ausgänge zu beeinflussen!

Wenn die Funktion aktiv ist (Meldung "Steuern fertig" am PG), können Sie einen NEUSTART oder einen MANUELLEN WIEDERANLAUF durchführen. Nach einem erneuten Übergang in den STOP können Sie wieder steuern. Die Ausgangssignalformer werden in diesem Fall **nicht gelöscht**.

Beenden der Funktion

Sie beenden die Funktion durch Drücken der Abbruchtaste am PG. Die Befehlsausgabesperre ist wieder aktiv (Leuchtdiode BASP = an).

Steuern Variablen

Mit der PG-Funktion "Steuern Variablen" können Sie die Werte von Operanden (Prozessvariablen) einmalig verändern. Dies ist in jedem Betriebszustand der CPU zulässig. Sie können alle Prozessvariablen angeben. Im Bereich des Prozessabbilds wird bei Zugriff auf eine Adresse, zu der keine Peripherie vorhanden ist, kein ADF ausgelöst.

Die Änderung wird an den Systemkontrollpunkten "asynchron", d.h. erst am Zyklusende, wirksam. Beachten Sie, dass die gesteuerten Werte nachträglich überschrieben werden können (z.B. durch das Anwenderprogramm oder die Prozessabbildaktualisierung)!

Hinweis!

Das PG steuert die Prozessvariablen E, A, M byteweise und DW, T, Z wortweise.
Wenn Sie **mehrere Operanden** steuern, dann werden die geänderten Bytes (bei DW, T, Z die Wörter) nacheinander, über mehrere Funktionsaufrufe verteilt, im Speicher geändert.

Tätigkeiten an Kontrollpunkten

Der nachfolgenden Tabelle können Sie entnehmen, welche Tätigkeiten bei den PG-Funktionen an den Kontrollpunkten ausgeführt werden.

Tätigkeiten der Online-Funktionen	Systemkontrollpunkt				Anwender-kontrollpunkt
	"STOP"	"ZYKLUS"	"WARTE-ZUSTAND"	"ASYN-CHRON"	
Eingabe der Adresse: Daten schreiben [1]	*		*	*	
Bausteineingabe: Baustein gültig erklären	*	*	*	*	
Baustein löschen	*		*	*	
Speicher komprimieren: Baustein verschieben [1] [2]	* [3]	*			
START/STOP	*	*	*	*	
URLÖSCHEN	*				
STATUS: Daten lesen und ausgeben					*
STATUS VARIABLEN: Daten lesen und ausgeben	*	*	*		
BEARBEITUNGSKONTROLLE Haltepunktvorgabe	*	*	*	*	
Daten lesen und ausgeben					*
STEUERN (Signalformer) [1]	*				
STEUERN VARIABLE [1]	*	*	*	*	

[1] Tätigkeiten, die über mehrere Systemkontrollpunkte verteilt werden können.

[2] Pro Systemkontrollpunkt maximal ein Baustein.

[3] Nach "Komprimieren im STOP" ist ein NEUSTART erforderlich.

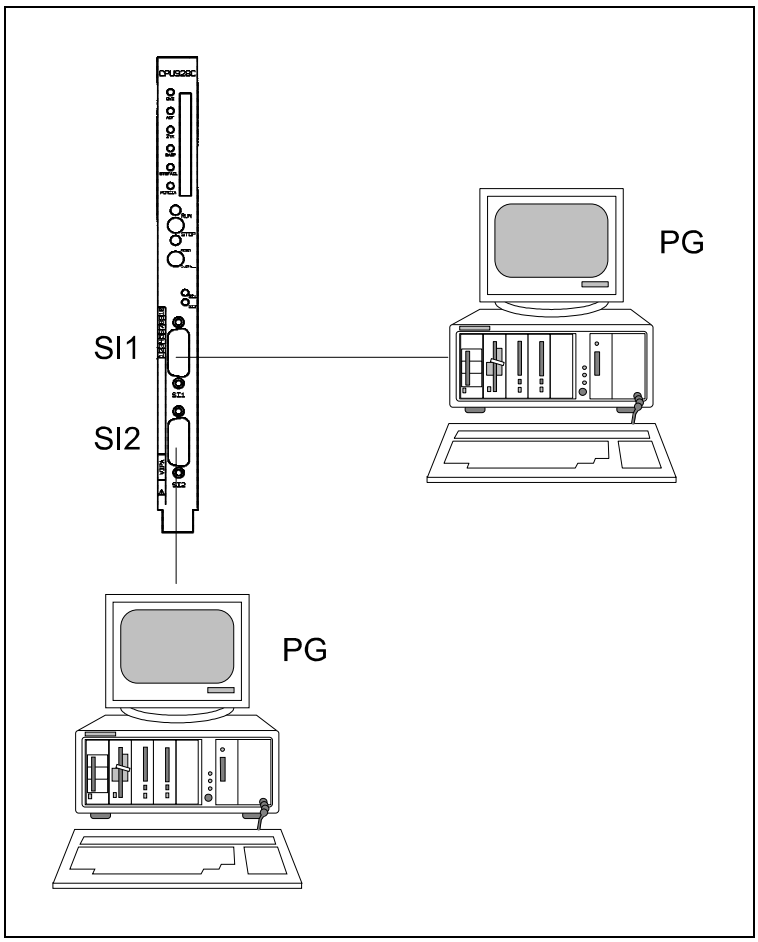
Serielle Kopplung

Für die serielle Kopplung PG-AG existieren folgende Möglichkeiten:

- Direkte Verbindung CPU - über die Standard-Steckleitung
- Verbindung zum PG über einen PG-Multiplexer
- Verbindung zum PG über Industrial Ethernet H1/L2/L1; dabei kann auch der PG-Multiplexer zwischengeschaltet werden.

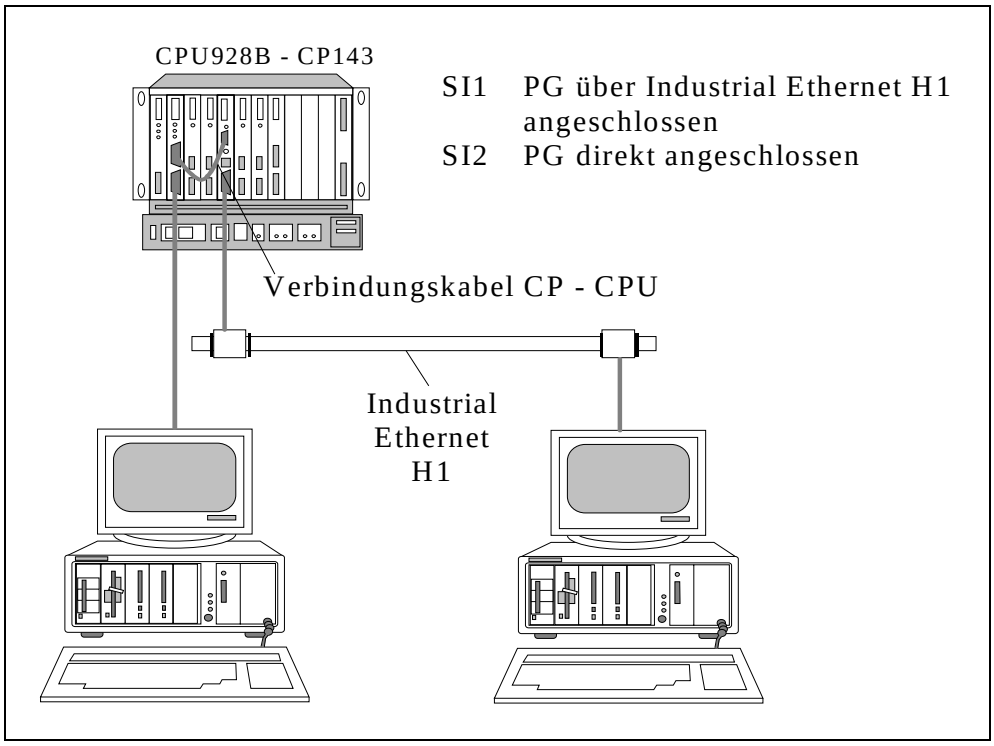
Parallelbetrieb von zwei seriellen PG-Schnittstellen

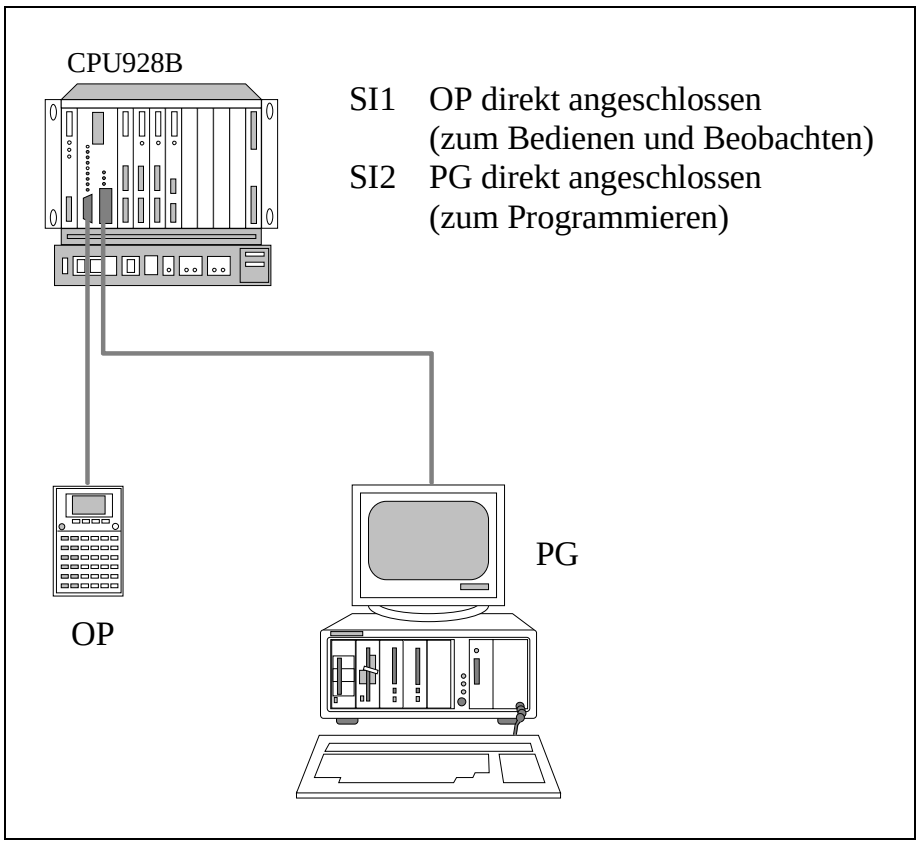
Die zweite Schnittstelle der CPU 928 können Sie ebenso wie die erste Schnittstelle als PG-Schnittstelle nutzen.



Sämtliche PG-Funktionen stehen Ihnen auf beiden Schnittstellen zur Verfügung. In den folgenden Abschnitten finden Sie nur die Informationen, die Sie benötigen, wenn Sie gleichzeitig an beiden Schnittstellen mit Programmiergeräten oder OPs arbeiten.

Konfigurations-Beispiele


CPU928B - CP143
SI1 PG über Industrial Ethernet H1 angeschlossen
SI2 PG direkt angeschlossen
Verbindungskabel CP - CPU
Industrial Ethernet H1



Inbetriebnahme

Um die zweite serielle Schnittstelle der CPU 928 als PG-Schnittstelle in Betrieb zu nehmen, schließen Sie das PG an die serielle Schnittstelle SI 2 an.

Hinweis!

Bitte beachten Sie, dass die CPU 928 standardmäßig mit einer 2. TTY-Schnittstelle ausgeliefert wird. Sie können die CPU 928 auch mit einer 2. RS 232C-Schnittstelle beziehen. Beide Schnittstellenarten besitzen die PG-Kommunikations-Funktionalität aber nur an die TTY-Schnittstelle können Sie ein PG direkt anschließen!

Betrieb

Wenn Sie die 2. serielle Schnittstelle als PG-Schnittstelle einsetzen, steht auf jeder Schnittstelle der volle Funktionsumfang der Standard-PG-Schnittstelle zur Verfügung. Dies gilt solange, wie die einzelnen Funktionen sich gegenseitig nicht beeinflussen, d.h. sequentiell nacheinander aufgerufen werden.

Zum Verständnis der Ausnahmen ist es sinnvoll, die PG-Funktionen in drei Gruppen zu unterteilen:

Gruppe	Kennzeichen
kurzlaufende Funktionen	Funktionen, die einen Auftrag ausführen und anschließend beendet sind. (z.B. "Übertragen", "Löschen" usw.)
langlaufende Funktionen	Funktionen, die eine feste Anzahl von Aufträgen bearbeiten: "Steuern", "Bearbeitungskontrolle".
zyklische Funktionen	Die Funktionen führen einen Auftrag immer wieder aus, bis sie vom Anwender beendet werden: "Status Baustein", "Status Variablen", "Steuern Variablen".

Achtung!

Bei langlaufenden und zyklischen Funktionen müssen Sie den Aufruf dieser Funktionen auf beiden PGs koordinieren.

Nicht parallel betreibbare Funktionen

In der nachfolgenden Tabelle sind die Funktionspaare aufgelistet, die Sie nicht parallel betreiben können.

Am ersten PG läuft Funktion:	Diese Funktion ist am zweiten PG *nicht möglich:*
"Steuern"	jede Funktion
"Bearbeitungskontrolle"	jede Funktion
eine "Status"-Funktion	"Steuern"
eine "Status"-Funktion	"Bearbeitungskontrolle"
eine "Status"-Funktion	"Urlöschen"
"Status" auf nicht bearbeiteten oder langlaufenden Baustein	jede Funktion

Wenn Sie dies missachten, meldet das zweite PG einen Fehler, z.B.:

"AS-Funktion gesperrt: laufende Funktion".

Die gleiche Fehlermeldung oder *"Zeitüberlauf bei Datenaustausch mit AG"* erscheint, wenn die CPU 928 gerade Funktionen des anderen PGs bearbeitet, welche den Zugriff Ihres PG innerhalb der Zeitüberwachung verhindern. Ihre Eingabe wird dann abgewiesen. Wiederholen Sie Ihre Eingabe, nachdem die Funktionen des anderen PGs abgeschlossen sind.

Hinweis!

Durch unterschiedlichen Leistungs- und Funktionsumfang sind Zeitüberwachung und Fehlermeldeverhalten nicht bei allen PGs und OPs gleich.

Achtung!

Wenn Sie gleichzeitig an beiden PGs die Funktion "Speicherausbau" aktivieren, kann es zu falschen Anzeigen kommen.

Wenn Sie gleichzeitig online an beiden PGs Bausteine eingeben, korrigieren oder löschen, so müssen Sie beachten, dass die Bausteine nicht vor dem Zugriff durch das jeweils andere PG geschützt sind.

"Status" auf nicht angesprungenen Baustein oder "Status" im STOP blockiert die andere Schnittstelle für alle Funktionen.

Ablauf bei bestimmten Betriebsfällen

Paralleler Betrieb bei kurzlaufenden Funktionen

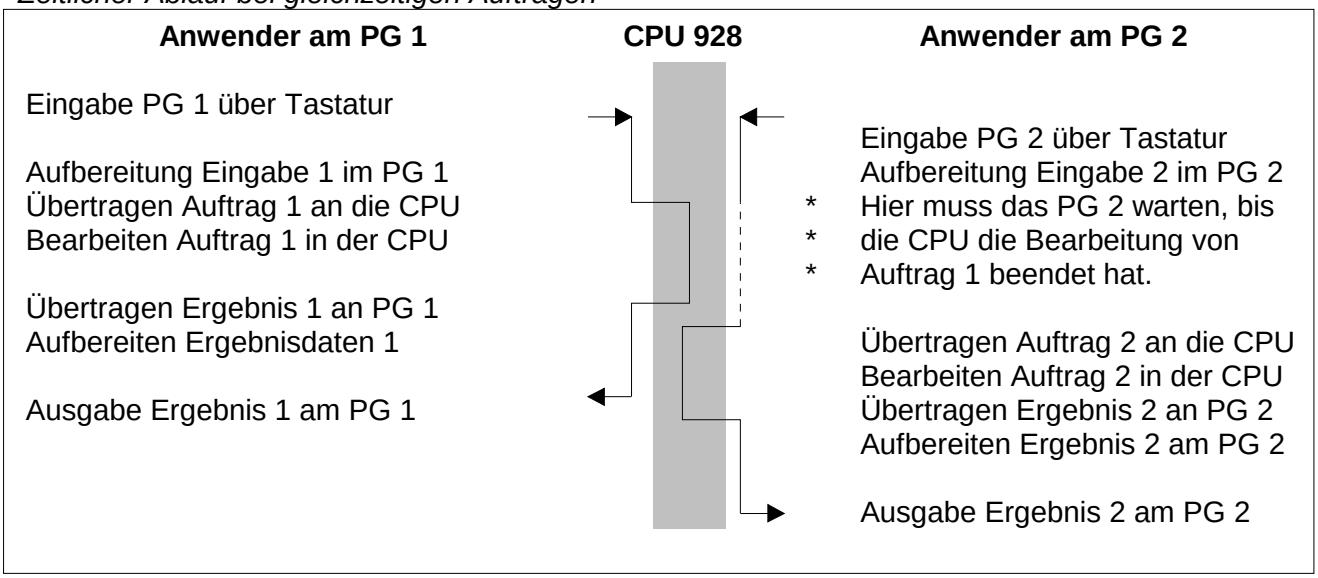
Wenn Sie an beiden Schnittstellen gleichzeitig mit Programmiergeräten arbeiten, versuchen beide PGs, unabhängig voneinander ihre jeweiligen Funktionen auszuführen. Solange sie ihre Aufträge zeitlich versetzt an die CPU übertragen, werden diese in der Reihenfolge ihres Eintreffens nacheinander bearbeitet.

Es kann aber vorkommen, dass die CPU 928 zwei Aufträge entweder gleichzeitig erhält oder einen Auftrag vom zweiten PG erhält, während noch ein Auftrag des ersten PGs läuft.

Da die gleichzeitige Bearbeitung nicht möglich ist, werden diese beiden Aufträge nacheinander bearbeitet; allerdings sind die Wartezeiten für den zweiten Auftrag meistens so gering, dass sie für den Anwender kaum in Erscheinung treten.

Es ergibt sich somit für gleichzeitige Aufträge folgender Ablauf:

Zeitlicher Ablauf bei gleichzeitigen Aufträgen


Anwender am PG 1
CPU 928
Anwender am PG 2
Eingabe PG 1 über Tastatur
Aufbereitung Eingabe 1 im PG 1
Übertragen Auftrag 1 an die CPU
Bearbeiten Auftrag 1 in der CPU
Übertragen Ergebnis 1 an PG 1
Aufbereiten Ergebnisdaten 1
Ausgabe Ergebnis 1 am PG 1
Eingabe PG 2 über Tastatur
Aufbereitung Eingabe 2 im PG 2
* Hier muss das PG 2 warten, bis
* die CPU die Bearbeitung von
* Auftrag 1 beendet hat.
Übertragen Auftrag 2 an die CPU
Bearbeiten Auftrag 2 in der CPU
Übertragen Ergebnis 2 an PG 2
Aufbereiten Ergebnis 2 am PG 2
Ausgabe Ergebnis 2 am PG 2

Aus diesem Ablauf ist zu ersehen, dass Sie zwar an beiden PGs unabhängig voneinander arbeiten können, dass aber eine gegenseitige Beeinflussung auftritt.

So kann es passieren, dass beide PG gleichzeitig denselben Baustein bearbeiten oder über ein PG ein Baustein gelöscht wird, den das andere PG gerade bearbeitet.

Sie müssen also bei jeder Bearbeitung berücksichtigen, inwieweit Sie das Verhalten des anderen PG beeinflussen.

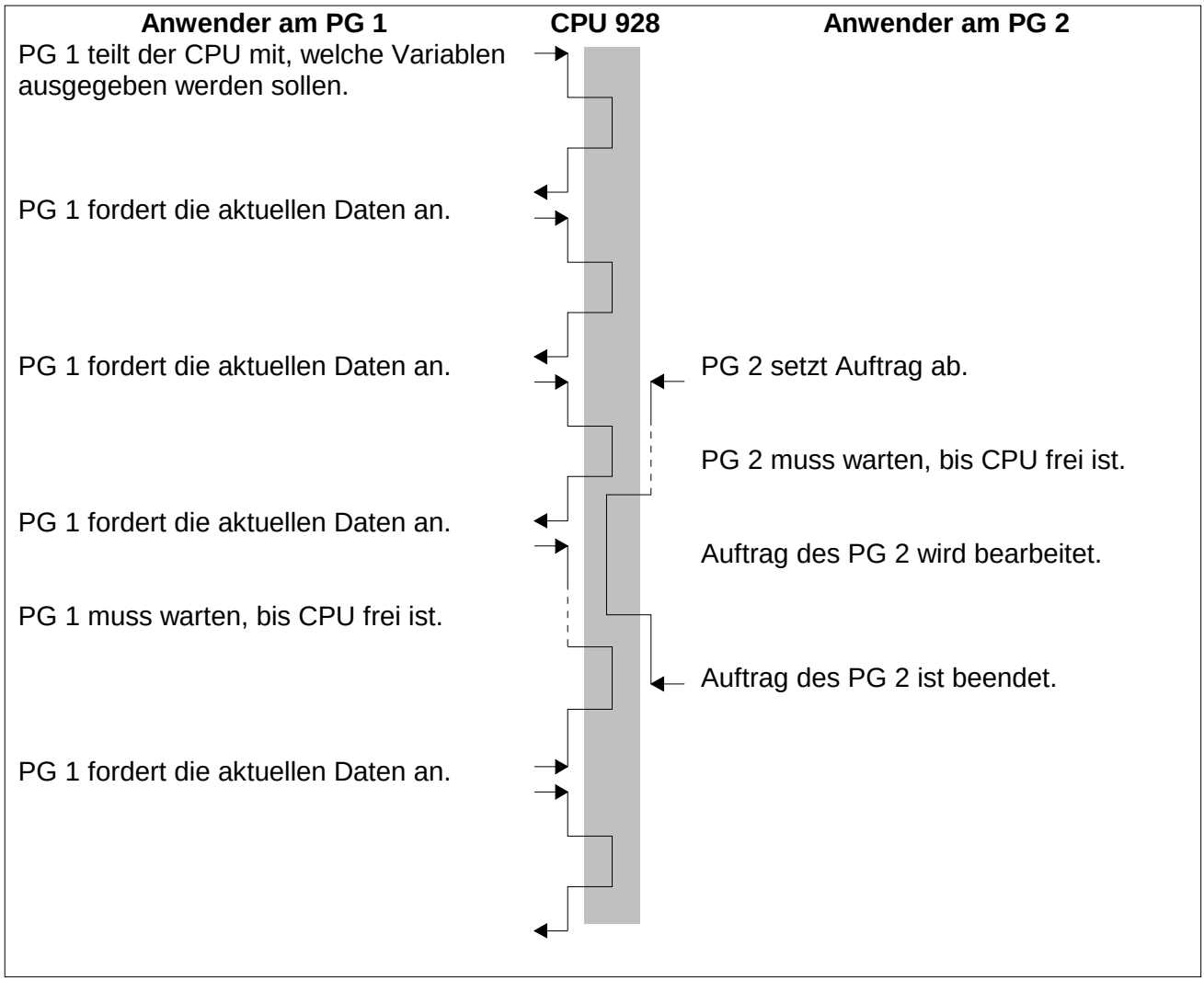
Paralleler Betrieb bei langlaufenden Funktionen

Die langlaufenden Funktionen "Steuern" und "Bearbeitungskontrolle" können keine andere Funktion unterbrechen und können auch von keiner anderen Funktion unterbrochen werden. Sie dürfen damit auch nicht parallel ausgeführt werden, d.h. sie werden wie ein Standardauftrag als ein Block behandelt.

Paralleler Betrieb bei zyklischen Funktionen

Zyklische Funktionen können parallel sowohl zu zyklischen als auch zu kurzlaufenden Funktionen ausgeführt werden. Als Beispiel wird hier der Standardablauf für die Funktion "Status Variablen" dargestellt.

Zeitlicher Ablauf einer zyklischen Funktion mit paralleler, kurzlaufender Funktion

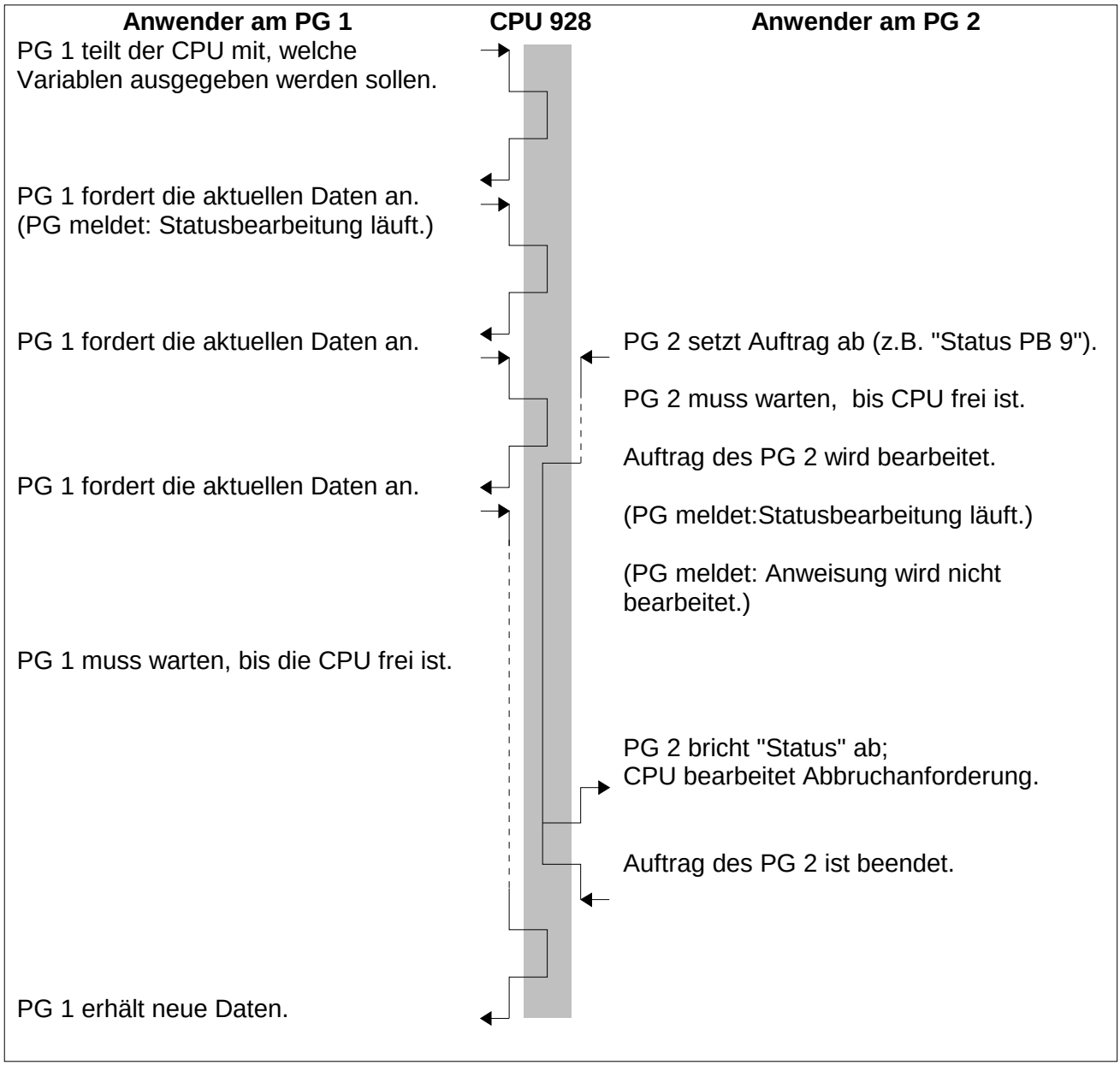

Anwender am PG 1
CPU 928
Anwender am PG 2
PG 1 teilt der CPU mit, welche Variablen ausgegeben werden sollen.
PG 1 fordert die aktuellen Daten an.
PG 1 fordert die aktuellen Daten an.
PG 2 setzt Auftrag ab.
PG 2 muss warten, bis CPU frei ist.
PG 1 fordert die aktuellen Daten an.
Auftrag des PG 2 wird bearbeitet.
PG 1 muss warten, bis CPU frei ist.
Auftrag des PG 2 ist beendet.
PG 1 fordert die aktuellen Daten an.

Um einem zweiten PG die Möglichkeit zu bieten, auch einen Auftrag an die CPU zu senden, wird die Statusfunktion zwischen zwei Anforderungen unterbrochen und nach dem eingeschobenen Auftrag wieder fortgesetzt. Da die unterbrechende Funktion zu ihrer Bearbeitung CPU-Leistung benötigt, muss die gesamte CPU-Systemleistung auf beide Funktionen aufgeteilt werden, z.B. werden bei der Funktion "Status Variablen" die auszugebenden Daten etwas langsamer aktualisiert.

Bei gleichzeitigem Arbeiten an beiden PGs ergibt sich der in der folgenden Abb. dargestellte Ablauf.

Dieser gilt auch für den Fall, dass auf beiden PGs zyklische Funktionen ablaufen; die beiden PGs wechseln sich dann beim Zugriff auf die CPU ab.

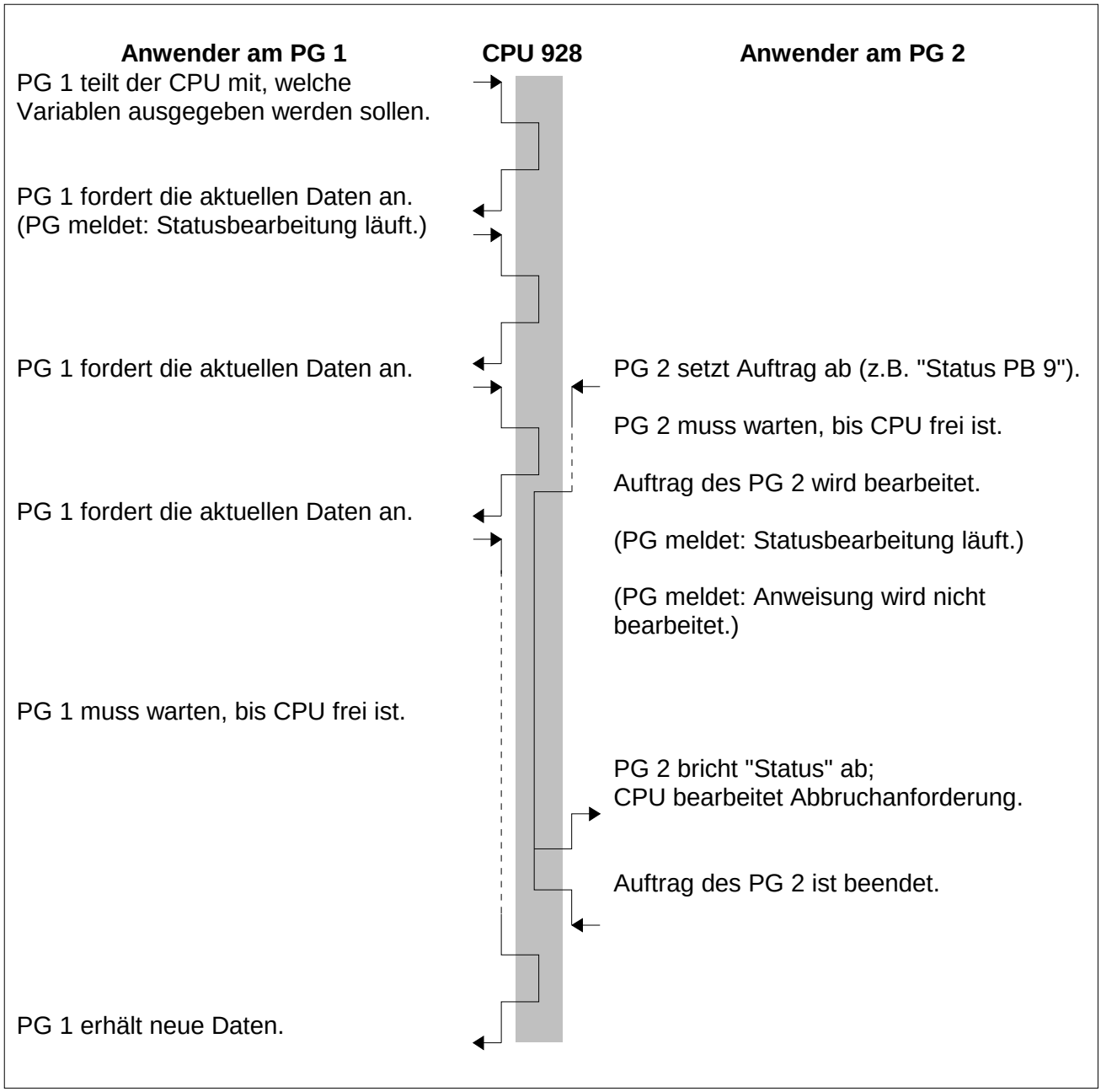
Zeitlicher Ablauf zweier paralleler zyklischer Funktionen


Anwender am PG 1
CPU 928
Anwender am PG 2
PG 1 teilt der CPU mit, welche Variablen ausgegeben werden sollen.
PG 1 fordert die aktuellen Daten an. (PG meldet: Statusbearbeitung läuft.)
PG 1 fordert die aktuellen Daten an.
PG 2 setzt Auftrag ab (z.B. "Status PB 9").
PG 2 muss warten, bis CPU frei ist.
Auftrag des PG 2 wird bearbeitet.
PG 1 fordert die aktuellen Daten an.
(PG meldet:Statusbearbeitung läuft.)
(PG meldet: Anweisung wird nicht bearbeitet.)
PG 1 muss warten, bis die CPU frei ist.
PG 2 bricht "Status" ab; CPU bearbeitet Abbruchanforderung.
Auftrag des PG 2 ist beendet.
PG 1 erhält neue Daten.

Besonderheiten bei zyklischen Funktionen auf beiden PGs

Blockiert die unterbrechende Funktion die CPU 928 ("Status" auf einen Baustein, der nicht ausgeführt wird), so wird die unterbrochene Funktion ebenfalls blockiert. Sie wird erst dann wieder fortgesetzt, wenn die unterbrechende Funktion beendet wird.

Bei gleichzeitigem Arbeiten an beiden PGs ergibt sich dann folgender Ablauf:


Anwender am PG 1
CPU 928
Anwender am PG 2
PG 1 teilt der CPU mit, welche Variablen ausgegeben werden sollen.
PG 1 fordert die aktuellen Daten an. (PG meldet: Statusbearbeitung läuft.)
PG 1 fordert die aktuellen Daten an.
PG 2 setzt Auftrag ab (z.B. "Status PB 9").
PG 2 muss warten, bis CPU frei ist.
Auftrag des PG 2 wird bearbeitet.
PG 1 fordert die aktuellen Daten an.
(PG meldet: Statusbearbeitung läuft.)
(PG meldet: Anweisung wird nicht bearbeitet.)
PG 1 muss warten, bis CPU frei ist.
PG 2 bricht "Status" ab; CPU bearbeitet Abbruchanforderung.
Auftrag des PG 2 ist beendet.
PG 1 erhält neue Daten.

Allgemeine Hinweise

Wird an einer Schnittstelle "Status Variablen", "Steuern Variablen" (bei der Statusanzeige) oder "Status" ausgegeben und an einer anderen Schnittstelle "Speicher komprimieren", "Baustein löschen" oder "Baustein übertragen" durchgeführt, so kann es zu einer falschen Anzeige in der Statusanzeige kommen.

Teil 12 Integrierte Sonderfunktionen

Einleitung

Das Betriebssystem der CPU 928 bietet Ihnen Sonderfunktionen an, die Sie bei Bedarf mit einem Bausteinaufruf ausführen können. Für diese Sonderfunktionen sind die Organisationsbausteine OB 100 bis OB 255 reserviert. Diese Funktionen werden als Integrierte Sonderfunktionen bezeichnet, da sie ein fester Bestandteil des Systemprogramms sind. Sie können als Anwender Sonderfunktionen nur ausführen. Sie können Sonderfunktionen weder lesen noch ändern.

Übersicht

In dem nachfolgenden Inhaltsverzeichnis finden Sie alle zur Zeit vorhandenen Sonderfunktionen der CPU 928.

Inhalt

Thema	Seite
Teil 12 Integrierte Sonderfunktionen	**12-1**
OB 110: Zugriff auf das Anzeigenbyte	12-5
OB 111: AKKU 1, 2, 3 und 4 löschen	12-7
OB 112/113: AKKU-Roll-Up/AKKU-Roll-Down	12-8
OB 120: "Alarme gemeinsam sperren" ein-/ausschalten	12-10
OB 121: "Weckalarme einzeln sperren" ein-/ausschalten	12-13
OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten	12-15
OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten	12-18
OB 134, OB 135, OB 136 und OB 139	12-20
OB 150: Systemzeit stellen/lesen	12-22
OB 151: Uhrzeitgesteuerten Weckalarm stellen/lesen	12-29
OB 152: Zyklusstatistik	12-36
OB 160 bis 163: Zählschleifen	12-43
OB 170: Bausteinstack (BSTACK) lesen	12-45
OB 180: Variabler Datenbaustein-Zugriff	12-50
OB 181: Datenbausteine (DB/DX) testen	12-54
OB 182: Datenbereich kopieren	12-56
OB 186: Speicher komprimieren	12-60
OB 190/192: Merker in Datenbaustein übertragen	12-62
OB 191/193: Datenblöcke in Merkerbereich übertragen	12-65
OB 216 bis 218: Kachelzugriffe	12-70
OB 216: Schreiben auf eine Kachel	12-71
OB 217: Lesen aus einer Kachel	12-74
OB 218: Belegung einer Kachel	12-76
OB 220: Vorzeichenerweiterung	12-78
OB 221: Zyklusüberwachung einstellen	12-79
OB 222: Zyklusüberwachungszeit neu starten	12-80

OB 228: Statusinformation einer Programmebene lesen 12-81
OB 230 bis 237: Funktionen für Standard-FBs 12-83
OB 250/251: Regelung /PID-Algorithmus 12-84
OB 250: PID-Algorithmus initialisieren 12-91
OB 251: PID-Alorithmus bearbeiten.....12-92
OB 254/255: Einen Datenbaustein verschieben/duplizieren.....12-98

Schnittstelle

Folgende Schnittstellen stehen Ihnen bei der Programmierung zur Verfügung:

Bausteinaufruf

Aufruf mit bedingtem/absolutem Bausteinaufruf SPB.../SPA...

Parameter

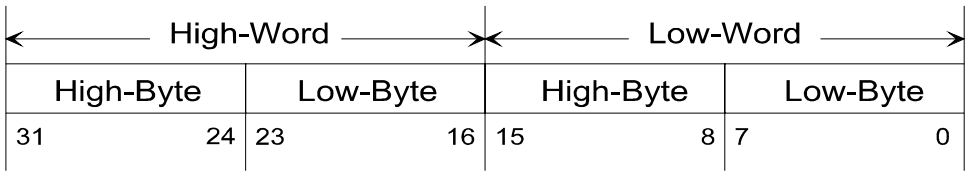
Parameter zur Voreinstellung über AKKU 1 und evtl. AKKU 2 und/oder Speicherzellen. Unter dem Begriff Parameter sind in der folgenden Beschreibung der einzelnen Sonderfunktionen alle Daten aufgelistet, die die CPU benötigt, um die Sonderfunktionen korrekt ausführen zu können.

Vor Aufruf der Sonderfunktionen im Anwender-Programm müssen Sie diese Daten in die Akkumulatoren laden.

Akku-Schreibweise

Bei den Angaben zur Parametrierung der einzelnen Sonderfunktions-Organisationsbausteine berücksichtigen Sie bitte folgende Schreibweise:

AKKU 1:	AKKU1	**32 bit**
AKKU-1-L:	AKKU1, Low-Wort	**16 bit**
AKKU-1-LL:	AKKU1, Low-Wort, Low-Byte	**8 bit**
AKKU-1-LH:	AKKU1, Low-Wort, High-Byte	**8 bit**



Fehlerbearbeitung

Tritt bei der Bearbeitung der aufgerufenen Sonderfunktion ein Fehler auf, so reagiert das Systemprogramm darauf mit einer speziellen Fehlerreaktion. Hierbei werden zwei Gruppen von Sonderfunktionen unterschieden:

- Fehler-OB, Akku-Anzeigen
- VKE, ANZ 0/ANZ1

Fehler-OB, Akku-Anzeigen

Es gibt Sonderfunktionen, bei denen im Fehlerfall ein Fehler-OB aufgerufen wird. In diesen können Sie das weitere Verhalten der CPU festlegen. Fehler-OBs sind die OBs 19, 30 und 31. In AKKU 1 und bei einigen Sonderfunktionen zusätzlich in AKKU 2 werden dem Fehler-OB Kennungen übergeben, die den aufgetretenen Fehler näher erläutern (siehe Kapitel "Unterbrechungs- und Fehlerbehandlung").

Stößt die CPU bei der Bearbeitung einer dieser Sonderfunktionen z.B. auf eine falsche Parametrierung, erkennt sie einen Laufzeitfehler und ruft den OB 31 auf. Ist z.B. die aufgerufene Sonderfunktion nicht vorhanden, erkennt die CPU einen Befehlscodefehler und versucht, den OB 30

aufzurufen. Wird bei einigen Sonderfunktionen in den Aufrufparametern auf einen Datenbaustein verwiesen und ist dieser Datenbaustein nicht geladen, versucht die CPU den OB 19 aufzurufen.

Falls die Fehler-OBs 30 bzw. 31 nicht geladen sind oder einen STP-Befehl enthalten, geht die CPU in den Stoppzustand. In den Steuerbits und im USTACK ist LZF bzw. BCF angekreuzt. In den AKKUs der Fehlerbearbeitungsebene sind Fehlerkennungen hinterlegt, die den Fehler näher beschreiben.

Falls die OBs 19, 30 oder 31 geladen sind (und keine STP-Operation enthalten), wird das Anwenderprogramm nach der Bearbeitung des betreffenden OBs mit dem nächsten Befehl fortgesetzt. In diesem Fall sind die AKKUs unverändert.

VKE, ANZ 0/ANZ1

Bei einigen Sonderfunktionen werden für das Anzeigen von sonderfunktionspezifischen Fehlern das VKE oder die Anzeigen ANZ 0/ANZ 1 beeinflusst.

Wenn bei der Bearbeitung dieser Sonderfunktion ein Fehler auftritt, wird in den meisten Fällen das VKE gesetzt (VKE=1). Sie können in Ihrem Anwender-Programm bei diesen Sonderfunktionen mit einer SPB-Operation (Springe bedingt) das VKE auswerten und damit auf einen Fehler reagieren.

Bei manchen Sonderfunktionen werden die Ergebnisanzeigen ANZ 0 und ANZ 1 durch die Bearbeitung der Sonderfunktion beeinflusst. Diese Anzeigen können Sie in Ihrem Anwender-Programm mit Vergleichsoperationen abfragen und somit ebenfalls auf einen Fehler reagieren.

Welche der geschilderten Fehlerreaktionen bei den einzelnen Sonderfunktions-OBs ausgeführt werden, wird in den folgenden Unterkapiteln zu den Sonderfunktions-OBs beschrieben.

Hinweis!

Der Aufruf eines Sonderfunktions-OB mit Operation "SPB OB" bzw. "SPA OB" wirkt nicht wie ein echter Bausteinwechsel, sondern wie eine SPS-Operation.
Es werden keine Alarme eingeschaltet (bei Voreinstellung "Unterbrechung an Bausteingrenzen")!

Sonderfunktionen mit Pseudobefehlsgrenzen

Einige der Sonderfunktionen sind langlaufende Sonderfunktionen, die sogenannte Pseudobefehlsgrenzen enthalten.

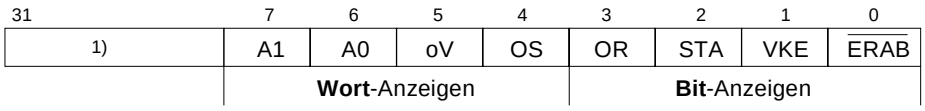
Dies bedeutet, dass die Ausführung der Sonderfunktion in mehreren Teilschritten erfolgt. Wenn nun während der Ausführung eines Teilschritts eine Unterbrechung (z.B. Weck- oder Prozessalarm an Befehlsgrenzen) auftritt, wird am Ende dieses Teilschritts an der Pseudobefehlsgrenze der entsprechende Organisationsbaustein eingeschachtelt.

OB 110: Zugriff auf das Anzeigenbyte

Funktion

Mit Hilfe des OB 110 können Sie das Anzeigenregister mit dem Inhalt des AKKU 1 beschreiben, mit "1"maskieren oder mit "0" maskieren.

Belegung des AKKU 1 für den Zugriff auf das Anzeigenregister:



1) Die Bits 8 bis 31 sind für Erweiterungen reserviert und müssen beim Beschreiben des Anzeigenregisters gleich '0' sein. Sie müssen beim Auslesen des Anzeigenregisters ignoriert werden.

Parameter

AKKU-2-L

Funktions-Nr.,

zulässige Werte: 1, 2 oder 3

AKKU 2

Neues Anzeigebyte bzw. Maske

Funkt.-Nr. in AKKU-1-L	Inhalt von AKKU-1-L		Funktion
	vorher	nachher	
1	neues Anzeigenbyte	neues Anzeigenbyte	Der Inhalt von AKKU 1 wird in das Anzeigenregister geladen
2	Maske	neues Anzeigenbyte	Alle in der Maske in AKKU 1 mit einer "1" gekennzeichneten Bits werden im Anzeigenregister gleich "1" gesetzt. Das neue Anzeigenbyte wird in AKKU 1 geladen.
3	Maske	neues Anzeigenbyte	Alle in der Maske in AKKU 1 mit einer "1" gekennzeichneten Bits werden im Anzeigenregister gleich "0" gesetzt. Das neue Anzeigenbyte wird in AKKU 1 geladen.

Ergebnis

Nach Ablauf des OB 110 ist das Anzeigenbyte entsprechend Funktion und AKKU-1 Inhalt verändert.

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- Funktions-Nr. in AKKU-2-L ungleich 1,2 oder 3
- In AKKU 1 ist eines der Bits Nr. 8 bis Nr. 31 gesetzt.

Im Fehlerfall wird der OB 31 (Sonstige Laufzeitfehler) aufgerufen. Ist der OB 31 nicht geladen, geht die CPU in den Stoppzustand. In beiden Fällen ist in AKKU-1-L die Fehlerkennung 1A49h hinterlegt.

Beispiel

Der OB 110 ist ein Hilfsmittel zum Testen der Operationen, die das Anzeigenregister auswerten oder beeinflussen. Seine Anwendung ist jedoch nicht hierauf begrenzt.

Folgendes Beispiel soll einen weiteren Anwendungsfall zeigen.

Aufrufverteiler:

In Abhängigkeit von dem Inhalt des Merkerbytes MB 0 soll eines von vier Teilprogrammen aufgerufen werden. Den vier Teilprogrammen werden die Bits M 0.0 bis M 0.3 zugeordnet.

Es darf immer nur eines dieser Bits gesetzt sein.

```
          :L      MB 0
          :SLW    4             M 0.0 bis M 0.3 um vier Bits
          :                     nach links schieben
          :L      KB 1          Funktions-Nr. laden
          :TAK
          :SPA    OB 110
          :SPS    =M000         Sprung, falls OS = 1
          :SPO    =M001         Sprung, falls OV = 1
          :SPM    =M002         Sprung, falls ANZ 0 =1
          :SPP    =M003         Sprung, falls ANZ 1 =1
          :
          :                     falls kein Bit gesetzt ist
          :
          :BEA
          :
M000      :                     falls M 0.0 = 1
          :
          :BEA
M001      :                     falls M 0.1 =1
          :
          :BEA
M002      :                     falls M 0.2 =1
          :
          :BEA
M003      :                     falls M 0.3 =1
          :BEA
```

OB 111: AKKU 1, 2, 3 und 4 löschen

Funktion Durch den einmaligen Aufruf des Sonderfunktions-Organisationsbausteins OB 111 können sie die Inhalte der AKKUs 1 bis 4 auf einfache Weise löschen: Der OB111 überschreibt alle vier Register mit '0'.

Parameter keine

Ergebnis Die AKKUs 1 bis 4 (je 32 Bit) sind gelöscht.

Fehlerfälle keine

OB 112/113: AKKU-Roll-Up/AKKU-Roll-Down

Funktion

OB 112 und OB 113 bewirken ein "Rollen" der AKKU-Inhalte in aufsteigender bzw. absteigender Richtung:

- Der OB 112 (Roll Up) verschiebt den Inhalt von AKKU 1 in den AKKU 2, den Inhalt von AKKU 2 in den AKKU 3 usw.
- Der OB 113 (Roll Down) verschiebt die AKKU-Inhalte in entgegengesetzte Richtung: Inhalt von AKKU 1 in den AKKU 4, AKKU 4 in AKKU 3 usw.

Parameter

keine

Ergebnis

Die folgenden Bilder zeigen die AKKU-Inhalte vor und nach dem Aufruf von OB 112 und OB 113.

Hinweis!

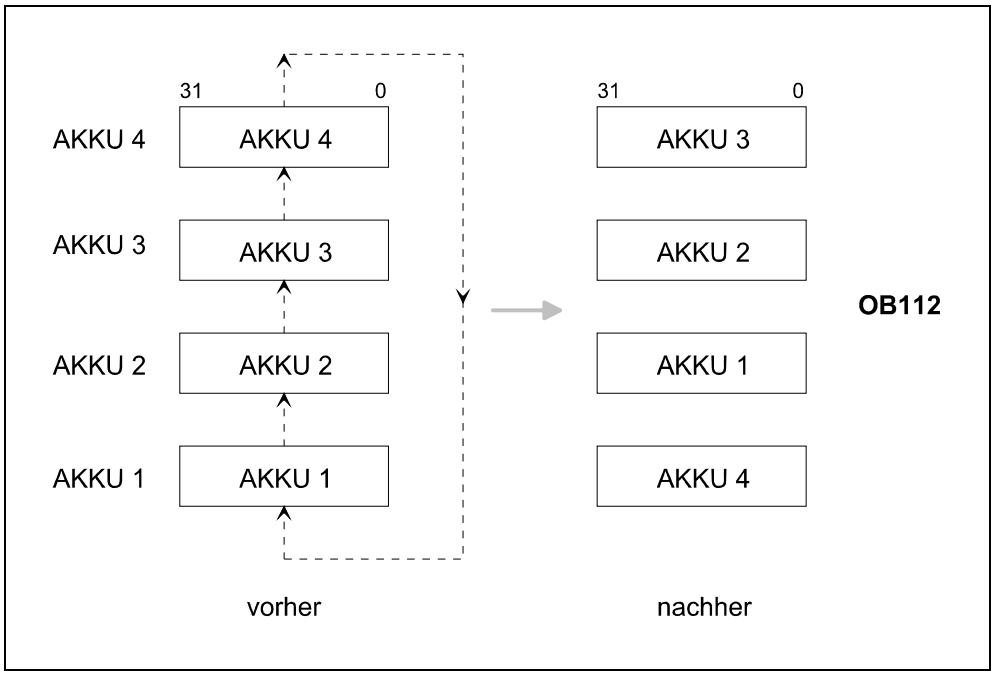
Mit den SPS-Operationen ENT (Ergänzender Operationsvorrat) und TAK (Systemoperation) lassen sich die AKKU-Inhalte ebenfalls verschieben

Fehlerfälle

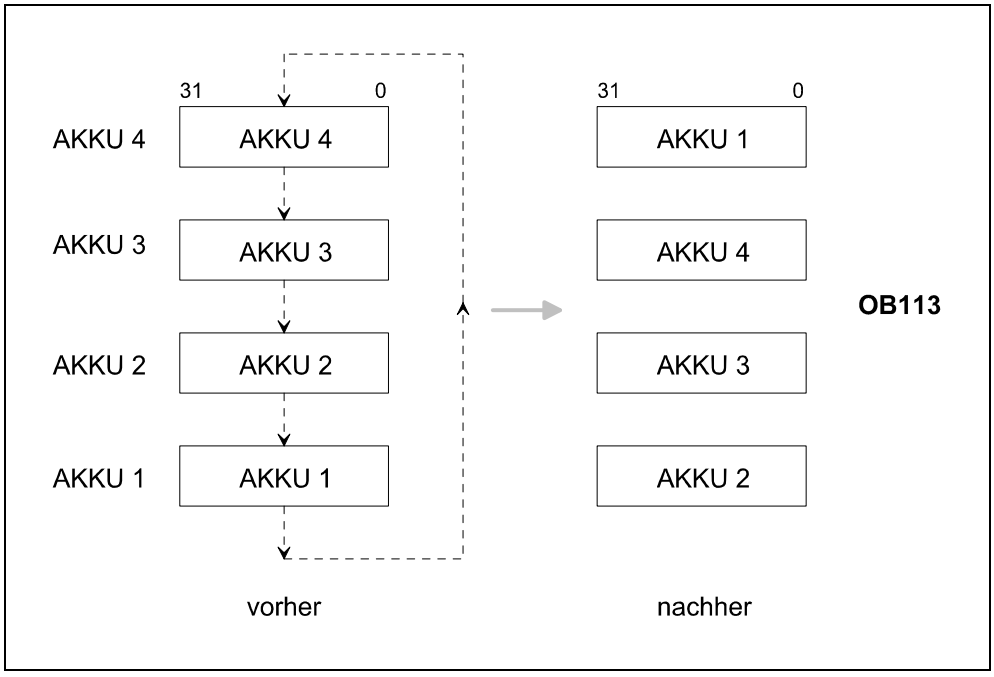
keine

Die folgenden Abbildungen zeigen Ihnen die Funktionsweise von "Roll-Up" bzw. "Roll-Down":

Wirkung der Funktion Roll Up



Wirkung der Funktion Roll Down



OB 120: "Alarme gemeinsam sperren" ein-/ausschalten

Einleitung

Ein SPS-Programm kann an Bausteingrenzen oder Befehlsgrenzen von Programmen höherer Priorität unterbrochen werden. Zu diesen höherprioren Programmbearbeitungsebenen gehören die Prozess- und alle Weckalarme, d.h. alle zyklischen, uhrzeitgesteuerten Weckalarme und Verzögerungsalarme.

Die Laufzeit des unterbrochenen Programms verlängert sich jeweils um die Laufzeit der eingeschachtelten Programme.

Mit Hilfe des OB 120 können Sie das Einschachteln von höherprioren Programmbearbeitungsebenen an einer oder mehreren aufeinanderfolgenden Baustein- oder Befehlsgrenzen (je nach Einstellungen im DX 0) verhindern.

Funktion

Der OB 120 reagiert auf alle Alarme:

"Alarme sperren" **einschalten** heißt, es werden ab sofort keine Alarme mehr registriert und diejenigen Alarme, die bereits registriert worden sind, die z.B. auf Bausteingrenzen "warten", werden gelöscht.

Nur falls der OB 2 (Prozessalarm) oder ein OB für die Weckalarmbearbeitung bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Alarme sperren" **ausschalten** heißt, es werden ab sofort alle an den Baustein- oder Befehlsgrenzen registrierten Alarme wieder eingeschachtelt und bearbeitet.

Parameter

1. Steuerdoppelwort

Der OB 120 speichert die zu sperrenden Alarme in einem systeminternen Steuerdoppelwort.

Die Bits des Steuerdoppelworts haben folgende Bedeutung:

Bit-Nr. im Steuerdoppelwort	Funktion
0 = '1'	alle zyklischen Weckalarme werden gesperrt
1 = '1'	der uhrzeitgesteuerte Weckalarm wird gesperrt
2 = '1'	alle Prozessalarme werden gesperrt
3 = '1'	der Verzögerungsalarm wird gesperrt
4 bis 31	reserviert; diese Bits müssen '0' sein!

Solange ein Bit auf '1' gesetzt ist, ist der entsprechende Alarm gesperrt.

2. Akkus

AKKU-2-L

Funktions-Nr.,

zulässige Werte: 1, 2 oder 3 mit:

1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.

2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

3: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

AKKU 1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion

Ergebnis

Der Aufruf des OB 120 bringt folgende Ergebnisse:

Funkt.-Nr. in AKKU-2-L	Inhalt von AKKU 1	
	vorher	nachher
1	Steuerwort	Steuerwort
2	Maske	neues Steuerwort
3	Maske	neues Steuerwort

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- unzulässige Funktions-Nr. in AKKU-2-L
- eines der reservierten Bits in AKKU 1 (Nr. 4 bis 31) ist auf '1' gesetzt.

Im Fehlerfall wird der OB 31 (sonstige Laufzeitfehler) aufgerufen. Ist der OB 31 nicht geladen, geht die CPU in den Stoppzustand. In beiden Fällen wird in AKKU-1-L die Fehlerkennung 1A47h hinterlegt.

Hinweise

Der Zustand des Steuerwortes lässt sich durch folgende Programmsequenz abfragen:

1. Funktions-Nr. 2 oder 3 in den AKKU-2-L laden,
2. Wert '0' in den AKKU 1 laden,
3. Sonderfunktions-OB 120 aufrufen,
4. AKKU 1 auslesen

- Der Zustand der Alarmverarbeitung kann auch durch Auslesen des Systemdatenworts BS 131 ermittelt werden.
  BS 131 Anzeigewort "Alarme gemeinsam sperren".

- Zum Sperren und Freigeben des Prozessalarms können statt des OB 120 die Befehle AS und AF verwendet werden:

  AS entspricht
  ```
  :L    KB 2
  :L    KM 00000000 00000100
  :SPA  OB 120
  ```

  AF entspricht
  ```
  :L    KB 3
  :L    KM 00000000 00000100
  :SPA  OB 120
  ```

OB 121: "Weckalarme einzeln sperren" ein-/ausschalten

Einleitung

Mit Hilfe des OB 121 können Sie das Einschachteln von bestimmten Weckalarm-OBs mit festem Zeitraster an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehlsgrenzen verhindern.

Z.B. können sie für einen bestimmten Programmteil festlegen, dass er nicht durch den OB 18 (5s) und den OB17 (2s) unterbrochen werden kann. Dagegen werden alle übrigen programmierten Weckalarme wie üblich bearbeitet.

Funktion

Der OB 121 hat Auswirkungen auf die Reaktion auf zyklische Weckalarme:

"Weckalarme einzeln sperren" einschalten heißt, es werden ab sofort keine der angegebenen zyklischen Weckalarme mehr registriert und diejenigen Alarme, die bereits registriert worden sind (die z.B. auf eine Bausteingrenze "warten"), werden gelöscht. Nur falls ein Weckalarm-OB zur Bearbeitung eines Weckalarms mit festem Zeitraster bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Weckalarme einzeln sperren" ausschalten heißt, es werden ab sofort alle auftretenden zyklischen Weckalarme wieder registriert, an der nächsten Baustein- oder Befehlsgrenze (je nach Einstellung im DX 0) eingeschachtelt und bearbeitet.

Parameter

1. Steuerwort

Der OB 121 vermerkt die zu sperrenden Weckalarme in einem systeminternen Steuerwort. Die Bits des Steuerwortes haben folgende Bedeutung:

Bit-Nr.	Alarm
0 bis 2	reserviert; diese Bits müssen '0' sein!
	Zyklische Weckalarme mit festem Zeitraster:
3 = '1'	10 ms (OB 10)
4 = '1'	20 ms (OB 11)
5 = '1'	50 ms (OB 12)
6 = '1'	100 ms (OB 13)
7 = '1'	200 ms (OB 14)
8 = '1'	500 ms (OB 15)
9 = '1'	1 s (OB 16)
10= '1'	2 s (OB 17)
11= '1'	5 s (OB 18)
12 bis 15	reserviert, diese Bits müssen '0' sein!

2. Akkus

AKKU-2-L

Funktions-Nr.

zulässige Werte: 1, 2 oder 3 mit:

1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.

2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

3: Alle in der Maske in AKKU 1 mit einer '1'gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

AKKU-1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion.

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- unzulässige Funktions-Nr. in den AKKU-2-L
- eines der reservierten Bits in den AKKU 1 ist gleich '1'

Im Fehlerfall wird der OB31 (Sonstige Laufzeitfehler) aufgerufen. Ist der OB31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen wird in den AKKU-1-L die Fehlerkennung 1A4Ah hinterlegt.

Hinweise

- Der Zustand des Steuerworts lässt sich durch folgende Programmsequenz abfragen:
  1. Funktions-Nr.2 oder 3 in den AKKU-2-L laden
  2. Wert '0' in den AKKU1 laden
  3. Sonderfunktions-OB121 aufrufen
  4. AKKU 1 auslesen

- Der Zustand der Alarmverarbeitung kann auch durch Auslesen des Systemdatenworts BS 135 ermittelt werden.

  BS135: Anzeigewort "Weckalarme einzeln sperren".

OB 122: "Alarme gemeinsam verzögern" ein-/ausschalten

Einleitung

Ein Anwender-Programm kann an Baustein- oder Befehlsgrenzen von Programmen höherer Priorität unterbrochen werden. Zu diesen höherprioren Programmbearbeitungsebenen gehören die Prozess- und alle Weckalarme (zyklische Weckalarme, uhrzeitgesteuerter Weckalarm und Verzögerungsalarm).

Die Laufzeit des unterbrochenen Programms verlängert sich dabei jeweils um die Laufzeit der eingeschachtelten Programme.

Mit Hilfe des OB 122 können Sie das Einschachteln von höherprioren Programmbearbeitungsebenen an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehlsgrenzen (je nach Einstellung im DX0) verhindern.

Funktion

Der OB 122 hat Auswirkung auf die Reaktion auf Alarme:

"Alarme verzögern" **einschalten** heißt, es werden weiterhin alle aufgetretenen Alarme registriert und bereits anstehenden Alarme bleiben registriert. Die registrierten Alarme werden nicht weiter bearbeitet. Vorübergehend werden alle Befehls- bzw. Bausteingrenzen für die Bearbeitung von Alarmen unwirksam gemacht. Nur falls der OB 2 (Prozessalarm) oder ein OB für die Weckalarmbearbeitung bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Alarme verzögern" **ausschalten** heißt, es werden alle registrierten Alarme an der nächsten Baustein- oder Befehlsgrenzen eingeschachtelt und bearbeitet.

Hinweis!

Wird innerhalb der "Alarme verzögern"- Phase ein bestimmter Weckalarm-OB zum zweiten Mal aufgerufen, kommt es zum Weckfehler.

Parameter

1. Steuerdoppelwort

Der OB 122 vermerkt die zu verzögernden Alarme in einem systeminternen Steuerdoppelwort.

Die Bits des Steuerdoppelwortes haben folgende Bedeutung:

Bit-Nr. im Steuerdoppelwort	Funktion
0='1'	alle zyklischen Weckalarme werden verzögert
1='1'	der uhrzeitgesteuerte Weckalarm wird verzögert
2='1'	alle Prozessalarme werden verzögert
3='1'	der Verzögerungsalarm wird verzögert
4 bis 31	reserviert; diese Bits müssen '0' sein!

Solange ein Bit auf '1' gesetzt ist, ist der betreffende Alarm verzögert.

2. Akkus

AKKU-2-L

Funktions-Nr.,

zulässige Werte: 1, 2 oder 3 mit:

- 1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.
- 2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.
- 3: Alle in der Maske in AKKU 1 mit einer '1'gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen. AKKU 1 neues Steuerwort oder Maske, abhängig von der gewünschten Funktion

Ergebnis

Der Aufruf des OB 122 bringt folgende Ergebnisse:

Funkt.Nr. in AKKU-2-L	Inhalt von AKKU 1	
	vorher	nachher
1	Steuerwort	Steuerwort
2	Maske	neues Steuerwort
3	Maske	neues Steuerwort

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- unzulässige Funktions-Nr. in den AKKU-2-L,
- eines der reservierten Bits in AKKU 1 (Nr. 4 bis 31) ist gleich '1'.

Achtung!

Im Fehlerfall wird der OB31 (sonstige Laufzeitfehler) aufgerufen. Ist der OB31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen wird in AKKU-1L die Fehlerkennung 1A48h hinterlegt.

Hinweise

- Der Zustand des Steuerwortes lässt sich durch folgende Programmsequenz abfragen:
  1. Funktions-Nr. 2 oder 3 in den AKKU-2-L laden
  2. Wert '0' in den AKKU 1 laden
  3. Sonderfunktions-OB122 aufrufen
  4. AKKU 1 auslesen

- Der Zustand der Alarmverarbeitung kann durch Auslesen von BS132 ermittelt werden.

  BS132: Anzeigewort "Alarme gemeinsam verzögern"

OB 123: "Weckalarme einzeln verzögern" ein-/ausschalten

Einleitung

Mit Hilfe des OB 123 können sie das Einschachteln von bestimmten Weckalarm-OBs (Weckalarme mit festem Zeitraster) an einer oder an mehreren aufeinanderfolgenden Baustein- oder Befehlsgrenzen verhindern.

Funktion

Der OB123 hat Auswirkung auf die Reaktion auf Alarme:

"Weckalarme einzeln verzögern" einschalten heißt, es werden weiterhin alle aufgetretenen Alarme registriert und bereits anstehenden Alarme bleiben registriert. Die registrierten Alarme werden nicht weiter bearbeitet. Vorübergehend werden an Befehls- bzw. Bausteingrenzen keine Alarme bearbeitet. Nur falls der Weckalarm-OB (zur Bearbeitung eines Weckalarms mit festem Zeitraster) bereits begonnen wurde, wird dieser vollständig bearbeitet.

"Weckalarme einzeln verzögern" ausschalten heißt, es werden ab sofort alle auftretenden zyklischen Weckalarme wieder registriert, an der nächsten Baustein- oder Befehlsgrenzen (je nach Einstellung im DX 0) eingeschachtelt und bearbeitet.

Hinweis!

Wird innerhalb der "Alarme verzögern"- Phase ein bestimmter Weckalarm-OB zum zweiten Mal aufgerufen, kommt es zum Weckfehler.

Parameter

1. Steuerdoppelwort

Der OB 123 vermerkt die zu verzögernden Alarme in einem systeminternen Steuerdoppelwort.

Die Bits des Steuerdoppelwortes haben folgende Bedeutung:

Bit-Nr.	Alarm
0 bis 2	reserviert, diese Bits müssen '0' sein!
	Zyklische Weckalarme mit festem Zeitraster:
3 ='1'	10 ms (OB 10)
4 ='1'	20 ms (OB 11)
5 ='1'	50 ms (OB 12)
6 ='1'	100 ms (OB 13)
7 ='1'	200 ms (OB 14)
8 ='1'	500 ms (OB 15)
9 ='1'	1 s (OB 16)
10='1'	2 s (OB 17)
11='1'	5 s (OB 18)
12 bis 15	reserviert, diese Bits müssen '0' sein!

2. Akkus

AKKU-2-L

Funktions-Nr.,

zulässige Werte: 1, 2 oder 3 mit:

1: Der Inhalt von AKKU 1 wird in das Steuerwort geladen.

2: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '1' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

3: Alle in der Maske in AKKU 1 mit einer '1' gekennzeichneten Bits werden im Steuerwort auf '0' gesetzt. Das neue Steuerwort wird in den AKKU 1 geladen.

AKKU 1

neues Steuerwort oder Maske, abhängig von der gewünschten Funktion.

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- unzulässige Funktions-Nr. in den AKKU-2-L,
- eines der reservierten Bits in AKKU 1 ist gleich '1'.

Achtung!

Im Fehlerfall wird der OB31 (Sonstige Laufzeitfehler) aufgerufen. Ist der OB31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen wird in AKKU-1L die Fehlerkennung 1A48h hinterlegt.

Hinweise

- Der Zustand des Steuerwortes lässt sich durch folgende Programmsequenz abfragen:
  1. Funktions-Nr. 2 oder 3 in den AKKU-2-L laden
  2. Wert '0' in den AKKU 1 laden
  3. Sonderfunktions-OB123 aufrufen
  4. AKKU 1 auslesen
- Der Zustand der Alarmverarbeitung kann durch Auslesen des Systemdatenworts BS137 ermittelt werden.

  BS137: Anzeigewort "Weckalarme einzeln verzögern".

OB 134, OB 135, OB 136 und OB 139

Rechenregister

Die Rechenregister werden durch eine arithmetische Operation (OB 134, 135, 136) wie folgt verändert (bei Festpunkt-Operationen nur das Low-Wort):

	AKKU 1	**AKKU 2**	**AKKU 3**	**AKKU 4**
vorher:	<AKKU 1>	<AKKU 2>	<AKKU3>	<AKKU 4>
nachher:	<Ergebn.>	<AKKU 3>	<AKKU 4>	<AKKU 4>

OB 134: *D

*D (Multipliziere 32-Bit-Festpunktzahlen) multipliziert AKKU 2 und AKKU 1 und lädt das Ergebnis in den AKKU 1.

Ist das Ergebnis größer als die größte oder kleiner als die kleinste darstellbare 32-Bit-Festpunktzahl, wird dies über OV = 1 und OS = 1 angezeigt. Danach werden AKKU 3 und AKKU 4 nach AKKU 2 und AKKU 3 übertragen.

.

Anzeigen

	A1	A0	OV	OS	OR	STA	VKE	/ER
hängt ab von	-	-	-	-	-	-	-	-
setzt	x	x	x	x	-	-	-	-

OB 135: :D

:D (Dividiere 32-Bit-Festpunktzahl) dividiert AKKU 2 und AKKU 1 und lädt das Ergebnis in den AKKU 1. Danach werden AKKU 3 und AKKU 4 nach AKKU 2 und AKKU 3 übertragen.

Anzeigen

	A1	A0	OV	OS	OR	STA	VKE	/ER
hängt ab von	-	-	-	-	-	-	-	-
setzt	x	x	x	x	-	-	-	-

OB 136: MOD

MOD (Rest der Division zweier 32-Bit-Festpunktzahlen) dividiert AKKU 2 und AKKU 1 und lädt als Ergebnis den Divisionsrest in den AKKU 1. Danach werden AKKU 3 und AKKU 4 nach AKKU 2 und AKKU 3 übertragen.

Anzeigen

	A1	A0	OV	OS	OR	STA	VKE	/ER
hängt ab von	-	-	-	-	-	-	-	-
setzt	x	x	x	x	-	-	-	-

OB 139: PUSH

PUHS (PUSH AKKU-Stack) trägt AKKU 1 tiefer in den AKKU-Stack ein.

PUSH kann dazu verwendet werden, um denselben Wert mehrfach in den AKKU-Stack einzutragen.

Anzeigen

	A1	A0	OV	OS	OR	STA	VKE	/ER
hängt ab von	-	-	-	-	-	-	-	-
setzt	-	-	-	-	-	-	-	-

OB 150: Systemzeit stellen/lesen

Eigenschaften der Systemzeit

- Die Auflösung beträgt 10ms beim Lesen und 1s beim Setzen.
- Schaltjahre werden berücksichtigt.
- Stundendarstellung wahlweise 24 Stunden oder 12 Stunden "am" (Vormittagszeit) und "pm" (Nachmittagszeit).
- Angabe des Wochentages
- Ein- und Ausgaben erfolgen BCD-codiert.
- Die integrierte Hardwareuhr wird über die Batterie im AG-Rahmen gepuffert. Wurde sie gestellt, so hat sie auch nach einem Spannungsausfall und Wiederanlauf den aktuellen Wert.

Funktion

Mit Hilfe des OB150 können Sie in Ihrem Anwenderprogramm Datum und Uhrzeit der CPU928 stellen oder auslesen. Datum und Uhrzeit werden im folgenden als "Systemzeit" bezeichnet.

Hinweis!

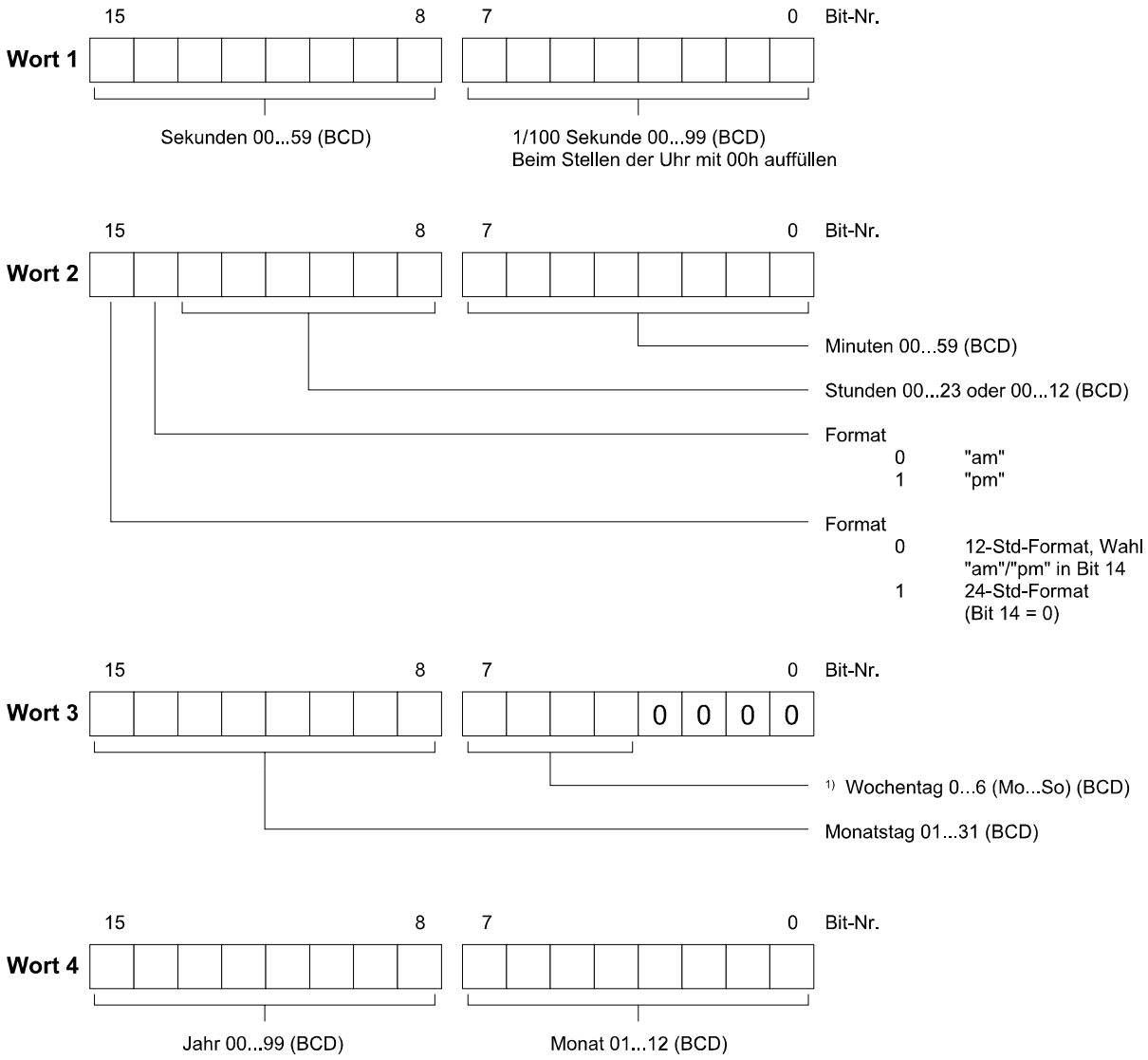
Bevor Sie die Systemzeit auslesen können, muss sie zunächst gestellt werden.

Parameter

1. Datenfeld für die Zeitparameter

Beim Stellen der Systemzeit übernimmt der OB 150 die Zeit aus einem Datenfeld, beim Lesen der Zeit überträgt der OB 150 die aktuellen Daten in das Datenfeld. Sie können dieses Datenfeld in einem Datenbaustein (DB) oder in einem der beiden Merkerbereiche (M oder S-Merker) anlegen.

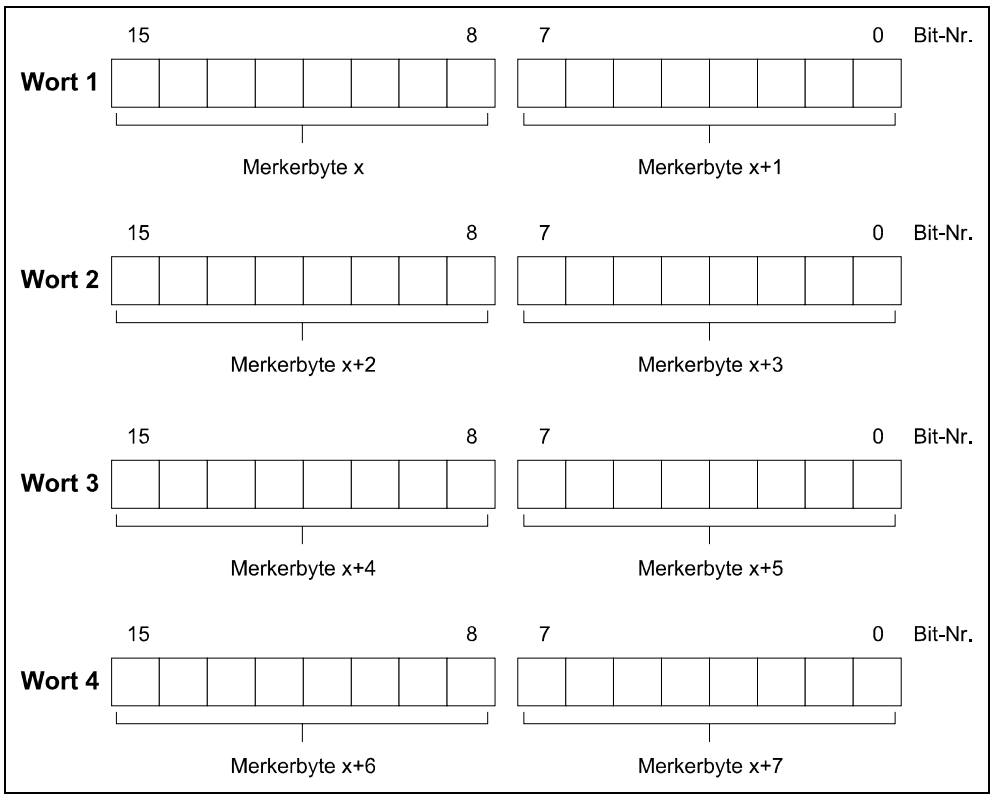
Datenfeld für Zeitparameter



1) Der angegebene Wert wird nach Aufruf von OB 150 logisch auf richtiges Datum unter Berücksichtigung der Schaltjahre überprüft.

Datenfeld im Merkerbereich für Zeitparameter

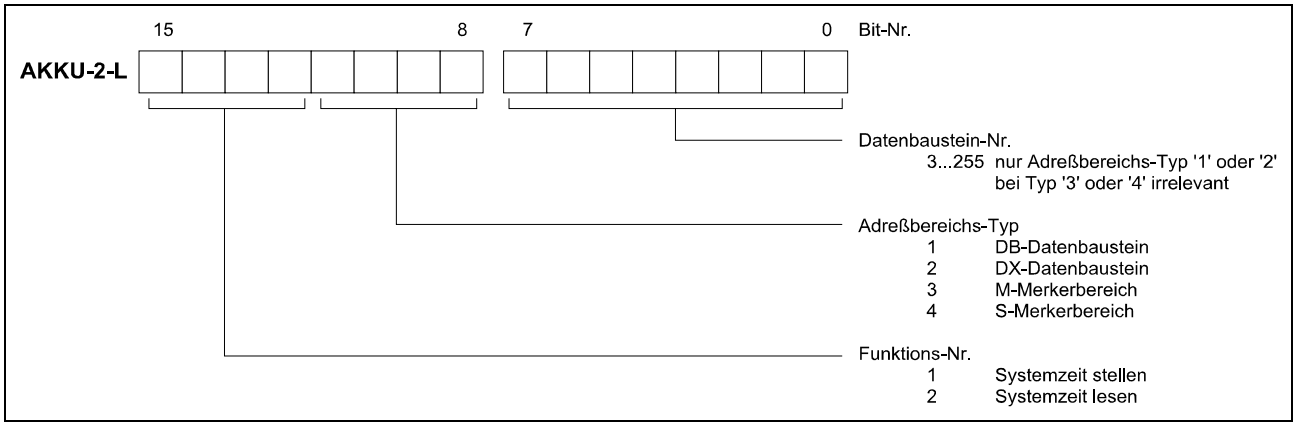
Wenn Sie das Datenfeld in einem Merkerbereich (M- oder S-Merker) anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1. Datenfeldwortes", den Sie beim Aufruf des OB 150 im AKKU-1-L hinterlegen müssen. Das nachfolgende Bild zeigt die Belegung der 4 Worte.



2. Akkus

AKKU-2-L

Der AKKU-2-L enthält Angaben zur gewünschten Funktion und zum verwendeten Datenfeld. Er muss folgenden Aufbau haben:



AKKU-1-L

Nr. des 1.Datenfeldwortes,

mögliche Werte (abhängig vom Adressbereichs-Typ):

DB, DX:	0 bis 2044
M-Merker	0 bis 248
	(=Nr. Merkerbyte 'x')
S-Merker	0 bis 1016
	(=Nr. Merkerbyte 'x').

Ergebnis

Nach korrekter Bearbeitung des OB 150 sind die Anzeigebits OR, ERAB und OS gelöscht, d.h. '0'. Alle anderen Anzeigebits sowie AKKU 1 und AKKU 2 sind unverändert.

Fehlerfälle

Im Fehlerfall werden der OB 19 oder OB 31 aufgerufen. Sind der OB 19 oder OB 31 nicht geladen, geht die CPU in den Stoppzustand. In beiden Fällen werden in AKKU1 und AKKU2 Fehlerkennungen hinterlegt (siehe nachfolgende Tabelle).

Fehlerkennungen des OB 150:

AKKU-1-L	AKKU-2-L	Bedeutung	aufgerufener OB
1A07h	-	DB nicht geladen	OB 19
1A4Ch	0001h	Funkt.-Nr. =0 oder >2	OB 31
	0100h	Adressbereichs-Typ unzulässig	
	0101h	DB-Nr. unzulässig	
	0102h	Nr. des 1. Datenfeldwortes unzulässig	
	0103h	DB-Länge im Bausteinkopf <5 Wörter	
	0201h	Jahresangabe im Datenfeld unzulässig	
	0202h	Monatsangabe im Datenfeld unzulässig	
	0203h	Monatstagangabe im Datenfeld unzul.	
	0204h	Wochentagangabe im Datenfeld unzul.	
	0205h	Stundenangabe im Datenfeld unzulässig	
	0206h	Minutenangabe im Datenfeld unzul.	
	0207h	Sekundenangabe im Datenfeld unzul.	
	0208h	1/100 Sekunde im Datenfeld ungleich 0	
	0209h	Datenfeldwort 3 BitNr. 0 bis 3 ≠ 0	
	020Ah	Stundenformat ungleich Einstellung bei OB 151	

Hinweis!

Wird bei der Funktion "Systemzeit stellen" eine fehlerhafte Parametrierung vorgenommen und es wurde vorher schon mindestens einmal die Uhrzeit korrekt gestellt, so führt dies zwar zu der geschilderten Übergabe der Fehlerkennung; die vorher eingestellte Systemzeit läuft jedoch weiter.

Beispiele

"Systemzeit stellen":

Die Systemzeit soll auf folgende Werte eingestellt werden:

Do, 24.10.1993, 11:30 Uhr 0 Sekunden, 24-Stunden-Darstellung

Es wird angenommen, dass die Zeitparameter im Datenbaustein DB 10 ab Datenwort DW 0 hinterlegt werden. Das Stellen der Systemzeit mit den so vorbereiteten Werten soll sekundengenau durch ein Eingangsbit (Triggerbit, z. B. E 1.0-Taster in der Nähe des AGs) ausgelöst werden.

Programmieren Sie zunächst den Datenbaustein DB 10 mit den nachfolgend aufgeführten Werten und laden Sie ihn in das AG. Die Operationen zum Aufrufen des OB 150 programmieren Sie im OB 1 (zyklische Bearbeitung). Programmieren Sie den OB 1 so, dass die Operationen zum Aufruf des OB 151 nur bei der kommenden Flanke des Triggerbits durchlaufen werden:

```
DB 10
      0: KH= 0000    linkes Byte = Sekunden (BCD),rechtes Byte=0
      1: KH= 9130    91 = Format (=80h) + Stunde (=11 BCD)
                     30 = Minuten (BCD)
      2: KH= 2430    24 = Monatstag (BCD)
                     30 = Wochentag (3=Donnerstag )+Bit0 bis 3=0
      3: KH= 9310    93 = Jahr (BCD)
                     10 = Monat (BCD)
```

Im OB 1 hinterlegte Operationen zum Aufruf des OB 150:

```
        :                 Signalflanke des Eingangs zum Stellen der
        :                 Systemzeit ist eingetroffen:
Stell   :L KH 1 1 0 A     Werte für AKKU-2-L:
        :       |         DB-Nr. = 10
        :       |         Adressbereichs-Typ=1 für "Datenfeld im DB"
        :       |         Funktions-Nr. = 1 für "Stellen"
        :L KF 0           AKKU-1-L:
        :                 Nr. 1. Datenfeldwort = 0
        :SPA OB 150
```

... Fortsetzung Beispiele

"Systemzeit auslesen":

Die aktuelle Systemzeit soll in den Datenbaustein DB 10 ab Datenwort DW 4 geschrieben werden. Dazu müssen Sie den OB 150 mit folgenden Parametern aufrufen:

```
:
:L KH 2 1 0 A      Werte für AKKU-2-L
:      | | |       DB-Nr. = 10
:      | |         Adressbereichs-Typ=1 für "Datenfeld im DB"
:      |           Funktions-Nr. = 2 für "Lesen"
:L KF 4            AKKU-1-L:
:                  Nr. 1. Datenfeldwort = 4
:SPA OB 150        OB 150 aufrufen
:A DB 10           DB 10 aufschlagen
:                  DB 10 auswerten
```

Nach dem Aufruf des OB 150 ist die aktuelle Systemzeit in folgender Form im DB 10 hinterlegt (Do, 24.10.93, 11:30 Uhr 20 Sekunden, 13 Hundertstel-Sekunden, 24-Stunden-Dastellung):

```
DW 4:KH=2013   Sekunden = 20 (BCD)
               1/100 Sekunden = 13 (BCD)
DW 5:KH=9130   Format = 24 Std. (Bit 15/14=10), Stunden=11 (BCD)
               Minuten = 30 (BCD)
DW 6:KH=2430   Monatstag = 24 (BCD)
               Wochentag = 3 =Donnerstag
DW 7:KH=9310   Jahr = 93 (BCD)
               Monat = 10 (BCD)
```

OB 151: Uhrzeitgesteuerten Weckalarm stellen/lesen

Funktion

Durch Aufruf des OB 151 können Sie:

- die CPU 928 dazu veranlassen, zu einer vorgegebenen Zeit den uhrzeitgesteuerten Weckalarm zu aktivieren:
  - minütlich
  - stündlich
  - täglich
  - wöchentlich
  - monatlich
  - jährlich
  - einmalig,
- den momentanen Status eines Zeitauftrags auslesen
- einen bereits erzeugten Zeitauftrag stornieren.

Der OB 151 kann in den Betriebszuständen ANLAUF und RUN aufgerufen werden. Ein erzeugter uhrzeitgesteuerter Weckalarm bleibt bei WIEDER-ANLAUF (automatisch oder manuell) erhalten. Bei Neustart wird ein vorhandener Zeitauftrag gelöscht.

Wenn Sie einen neuen Zeitauftrag erzeugen, wird ein laufender automatisch storniert. Es ist also immer nur ein uhrzeitgesteuerter Weckalarm aktiv.

Parameter

1. Datenfeld für die Auftragsparameter

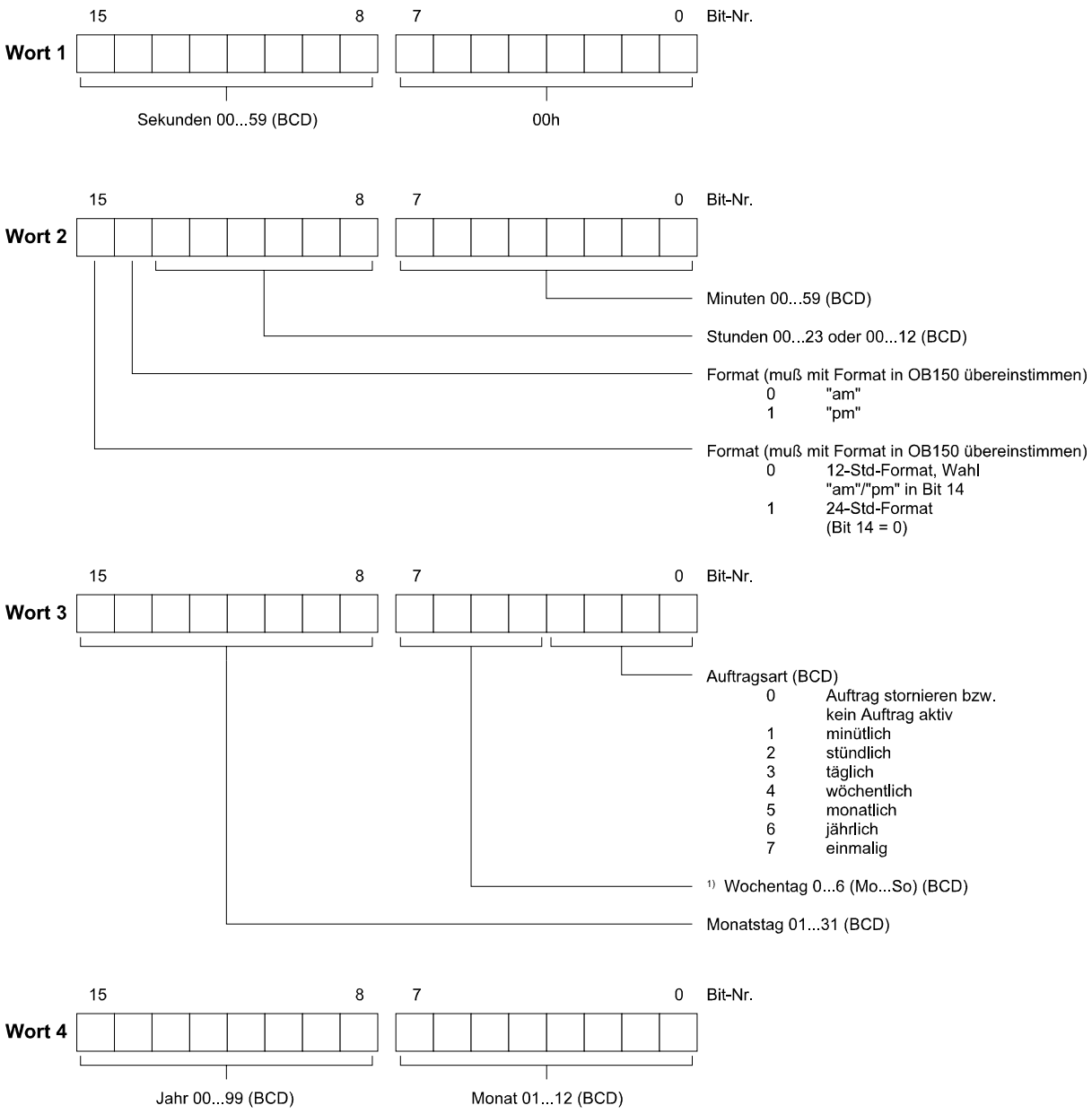
Beim **Erzeugen** bzw. **Stornieren** eines Zeitauftrags entnimmt der OB 151 die benötigten Auftragsparameter einem Datenfeld.

Beim **Auslesen** des aktuellen Zustands der Auftragsverwaltung überträgt der OB 151 die aktuellen Auftragsparameter in ein Datenfeld.

Sie können dieses Datenfeld in einem **Datenbaustein** oder in einem der beiden **Merkerbereiche** (M- oder S-Merker) anlegen.

Das Datenfeld besteht aus vier Wörtern und hat beim Erzeugen und Auslesen eines Zeitautrags folgendes Format:

Datenfeld für Auftragsparameter

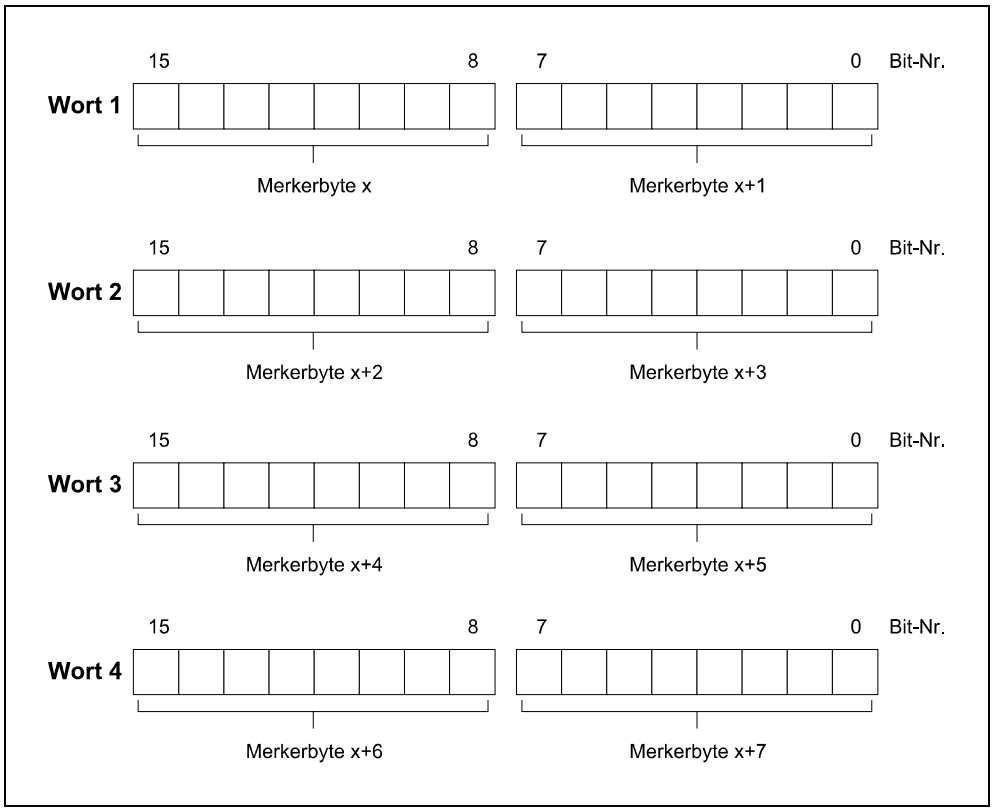


1) Der angegebene Wert wird nach Aufruf von OB 150 logisch auf richtiges Datum unter Berücksichtigung der Schaltjahre überprüft.

Datenfeld im Merkerbereich

Wenn Sie das Datenfeld in einem Merkerbereich (M- oder S-Merker) anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1.Datenfeldwortes", den Sie beim Aufruf des OB 150 im AKKU-1-L hinterlegen müssen:

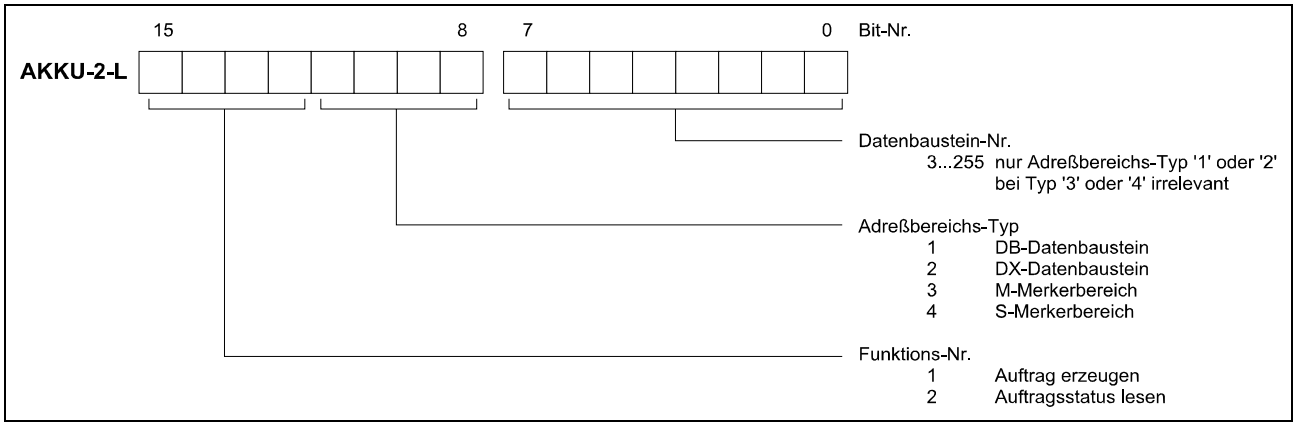
Datenfeld im Merkerbereich für Auftragsparameter



AKKUs

AKKU-2-L

Der AKKU-2-L enthält Angaben zur gewünschten Funktion und zum verwendeten Datenfeld. AKKU-2-L hat folgenden Aufbau:



AKKU-1-L

Nummer des 1. Datenfeldwertes,

mögliche Werte (in Abhängigkeit vom Adressbereichstyp):

DB, DX	0 bis 2044
M-Merker:	0 bis 248
	(=Nr. Merkerbyte 'x')
S-Merker	0 bis 1016
	(=Nr. Merkerbyte 'x').

Ergebnis

Nach korrekter Bearbeitung des OB 151 sind die Anzeigebits OR, ERAB und OS gelöscht (= '0').

Alle anderen Anzeigebits sowie AKKU 1 und AKKU 2 sind unverändert.

Hinweis!

Wenn beim Auslesen des Zeitauftrags im Datenfeld die Auftragsart '0' und alle restlichen Parameter 'F' oder 'FF' (hexadezimal) sind, ist kein Zeitauftrag aktiv.

- Dieser Zustand kann entstehen
- wenn NEUSTART durchgeführt wurde, ohne einen Zeitauftrag zu erzeugen,
- wenn ein einmaliger Zeitauftrag fällig war
- wenn Sie einen Auftrag storniert haben.

Achtung!

Fehlerfälle:

Im Fehlerfall werden der OB 19 oder der OB 31 aufgerufen. Sind der OB 19 oder der OB 31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen werden in den AKKU 1 und AKKU 2 Fehlerkennungen hinterlegt (siehe Tabelle).

Fehlerkennungen des OB 151:

OB 151 Fehlerkennungen

AKKU-1-L	AKKU-2-L	Bedeutung	aufgerufener OB
1A07h	-	DB nicht geladen	OB 19
1A4Dh	0001h	Funkt.-Nr. =0 oder >2	OB 31
	0100h	Adressbereichs-Typ unzulässig	
	0101h	DB-Nr. unzulässig	
	0102h	Nr. 1. Datenfeldwort unzulässig	
	0103h	DB-Länge im Bausteinkopf <5 Wörter	
	0201h	Jahresangabe im Datenfeld unzulässig	
	0202h	Monatsangabe im Datenfeld unzulässig	
	0203h	Monatstagangabe im Datenfeld unzulässig.	
	0204h	Wochentagangabe im Datenfeld unzul.	
	0205h	Stundenangabe im Datenfeld unzulässig	
	0206h	Minutenangabe im Datenfeld unzulässig	
	0207h	Sekundenangabe im Datenfeld unzulässig.	
	0208h	1/100 Sekunde im Datenfeld ungleich 0	
	0209h	Auftragsart im Datenfeld >7	
	020Ah	Stundenformat ungleich Einst. OB 150	

Hinweis!

Wird eine fehlerhafte Parametrierung vorgenommen und es wurde vorher schon ein gültiger Auftrag erzeugt, so führt dies zwar zu der geschilderten Übergabe der Fehlerkennungen; der vorher erzeugte Zeitauftrag bleibt jedoch weiter bestehen.

Was Sie bei den Zeitparametern beachten müssen

Abhängig davon wann ein Zeitauftrag ausgelöst werden soll, sind die einzelnen Zeitparameter in bestimmten Kombinationen vorzugeben. Hierbei müssen manche Parameter vorgegeben werden, andere werden vom Systemprogramm nicht ausgewertet und können daher entfallen.

Nachfolgende Tabelle gibt Ihnen an, welche Zeitparameter bei welchem Zeitauftrag angegeben werden müssen:

XXX = notwendige Angabe
--- = irrelevant

Zuordnung "Zeitauftrag - Zeitparameter"

Weckzeit	Sekunden	Minuten	Stunden	Wochen-tag	Monats-tag	Monat	Jahr
minütlich	XXX	---	---	---	---	---	---
stündlich	XXX	XXX	---	---	---	---	---
täglich	XXX	XXX	XXX	---	---	---	---
wöchentl.	XXX	XXX	XXX	XXX	---	---	---
monatlich	XXX	XXX	XXX	---	XXX	---	---
jährlich	XXX	XXX	XXX	---	XXX	XXX	---
einmalig	XXX	XXX	XXX	---	XXX	XXX	XXX

Besonderheiten

- Wenn bei Auftragsart "jährlich" (=6) der "29. Februar" als Monatstag und Monat gewählt wird, so resultiert daraus, dass der OB 9 nur in jedem Schaltjahr aufgerufen wird.
- Wenn bei Auftragsart "monatlich" (=5) als Monatstag der Wert "29", "30" oder "31" gewählt wird, so wird der OB 9 nur in den Monaten aufgerufen, bei denen dieser Tag vorhanden ist.

Beispiele

Verschiedene Zeitaufträge (im 24-Std.-Anzeigeformat):

1. "Auftrag jede Minute beim Sekundenwert 29" (12:44:29, 12:45:29 usw.):
   Sie müssen angeben: Auftragsart = 1 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 29
2. "Auftrag Stündlich jeweils um xx:14:15 Uhr":
   Sie müssen angeben: Auftragsart = 2 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 15
   Minuten = 14
3. "Auftrag täglich um 5:32:47 Uhr":
   Sie müssen angeben: Auftragsart = 3 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 47
   Minuten = 32
   Stunden = 05
4. "Auftrag wöchentlich, Dienstags um 10:50:00 Uhr":
   Sie müssen angeben: Auftragsart = 4 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 00
   Minuten = 50
   Stunden = 10
   Wochentag = 01
5. "Auftrag monatlich, jeweils am 14. Um 7:30:15 Uhr":
   Sie müssen angeben: Auftragsart = 5 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 15
   Minuten = 30
   Stunden = 07
   Monatstag = 14
6. "Auftrag jährlich, jeweils am 1. 5. um 00:01:45 Uhr":
   Sie müssen angeben: Auftragsart = 6 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 45
   Minuten = 01
   Stunden = 00
   Monatstag = 01
   Monat = 05
7. "Auftrag einmalig am 31.12.1999 um 23:55:00 Uhr":
   Sie müssen angeben: Auftragsart = 7 (Funktions-Nr. in AKKU-2-L =1)
   Sekunden = 00
   Minuten = 55
   Stunden = 23
   Monatstag = 31
   Monat = 12
   Jahr = 99
8. "Auftrag stornieren":
   Sie müssen angeben: Auftragsart = 0 (Funktions-Nr. in AKKU-2-L =1)
9. "Zeitauftrag auslesen":
   Sie müssen angeben: Funktions-Nr. in AKKU-2-L =2
   Falls kein Auftrag aktiv ist erhalten Sie als Ergebnis im Datenfeld:
   Datenfeldwort 0: FFFFh
   Datenfeldwort 1: FFFFh
   Datenfeldwort 2: FFF0h
   Datenfeldwort 3: FFFFh

OB 152: Zyklusstatistik

Einleitung

In der CPU 928 können eine Reihe statistischer Daten zur Zyklusdauer erfasst werden (Zyklusstatistik). Mit Hilfe des OB 152 können Sie die Zyklusstatistik initialisieren, die Statistikdaten auslesen und das Erfassen der Statistikdaten ein- und ausschalten.

Übersicht

Die statistischen Daten umfassen:

- die Dauer des vorangegangenen Zyklus,
- die seit der letzten Zyklusgrenze im aktuellen bearbeiteten Zyklus verstrichene Zykluszeit,
- die minimale und die maximale Zykluszeit seit der letzten Initialisierung der Zyklusstatistik,
- die Anzahl der Zyklen, die seit der letzten Initialisierung der Zyklusstatistik in die Statistik aufgenommen wurden,
- den Mittelwert der Zykluszeit: Es werden maximal die letzten 256 in der Statistik erfassten Zyklen zur Berechnung des Mittelwertes verwendet.

Hinweis!

In der Zyklusstatistik werden nur "normale" Zyklen erfasst. Wurde das Erfassen der Dauer des aktuellen Zyklus z.B. durch einen WIEDERANLAUF verfälscht, so werden die Daten dieses Zykluses nicht in die Statistik übernommen. Dadurch wirken sich "Ausreißer" nicht auf die Statistik aus.

Zykluszeiten, die länger als 167s sind, führen zu fehlerhaften Einträgen.

Ein-/Ausschalten der Statistik

Nach einem NEUSTART (automatisch oder manuell) ist die Statistik-Funktion immer ausgeschaltet und die statistischen Daten sind gelöscht (die Zyklusstatistik ist initialisiert). Ein WIEDERANLAUF (automatisch oder manuell) verändert weder den Zustand der Statistik noch die statistischen Daten.

Sie können die Statistikfunktionen im Betriebszustand ANLAUF oder RUN mit Hilfe des OB 152 einschalten.

Wurde die Statistik-Funktion mit dem OB152 eingeschaltet, werden die statistischen Daten an jeder Zyklusgrenze aktualisiert und können über Aufruf des OB 152 ausgelesen werden.

Wenn Sie die Statistikfunktionen nicht mehr benötigen, können Sie sie im Betriebszustand ANLAUF oder RUN über den OB 152 wieder ausschalten. Dadurch wird die Belastung der Zykluszeit, die durch das Aktualisieren der Zyklusdaten an jeder Zyklusgrenze entsteht, klein gehalten.

Initialisieren können Sie die Zyklusstatistik ebenfalls über den OB 152 im ANLAUF oder im RUN.

Möglich wäre es z.B. nach Auswertung der statistischen Daten (u.U. in Abhängigkeit vom Wert des Zyklus-Zählers) die Zyklusstatistik zu initialisieren.

Statistik-Daten

Die Statistik-Daten werden als einzelne Größen entweder direkt über den OB 152 ausgelesen oder beim Aufruf des OB 152 berechnet. Sie werden vom OB152 in AKKU-1-L bzw. AKKU-2-L übergeben.

Folgende Statistik-Größen können Sie durch Aufruf des OB 152 ermitteln:

Funktion-Nr.	Statistik-Größe	Bedeutung	Format	Einheit	Werte-bereich
1, 5	AKTZYK	bereits abgelaufene Zeit des akt. Zyklus	Hexa-dezimal	2)	2)
1, 5	LASTZYK	Dauer des letzten abgelaufenen Zyklus	Hexa-dezimal	2)	2)
2, 6	MINZYK	Dauer des kürzesten Zyklus seit der letzten Initialisierung	Hexa-dezimal	2)	2)
2, 6	MAXZYK	Dauer des längsten Zyklus seit der letzten Initialisierung	Hexa-dezimal	2)	2)
3, 7	MITTEL-WERT	Mittelwert der Zykluszeiten der zuletzt abgelaufenen - max. 256 - Zyklen. 1)	Hexa-dezimal	2)	2)
3, 7	ZYKLUS-ZÄHLER	Anzahl der seit der letzten Initialisierung der Zyklusstatistik aufgenommen. Zyklen	Hexa-dezimal	Anzahl der Zyklen	0 bis 0FFFFh
9	10-µs	kontinuierlich laufender Zähler	Hexa-dezimal	10µs	3)

1) Siehe "Berechnung des Mittelwertes"

2) Bei den Funktions-Nr. 1, 2 und 3 ist die Einheit ms, und der Wertebereich geht von 0 bis 0FFFFh (im Low-Word der Akkus),
Bei den Funktions-Nr. 5, 6 und 7 ist die Einheit 10 µs, und der Wertebereich geht von 0 bis 0FF FFFFh (im High- und Low-Word der Akkus)

3) Der 10-µs-ZÄHLER wird als 8-stelliger Hexadezimalwert im High- und Low-Word von Akku 1 ausgegeben

Berechnung des Mittelwertes

Der Mittelwert wird durch den OB 152 nach folgendem Algorithmus berechnet:

In einem systeminternen Umlaufpuffer wird bei jedem Aktualisieren der statistischen Daten der Wert von LASTZYK gespeichert. Dieser Puffer kann maximal 256 Werte aufnehmen. Ist der Puffer gefüllt, fällt der jeweils älteste Wert von LASTZYK heraus und der neueste Wert wird abgespeichert. Zusätzlich wird bei jeder Aktualisierung der Daten der Summenwert von den im Puffer enthaltenen LASTZYK-Werten so gebildet, dass immer die neuesten LASTZYK-Werte (maximal 256) darin enthalten sind. Bei Aufruf des OB 152 wird der Mittelwert gebildet durch Division des Summenwertes durch die Anzahl LASTZYK-Werte, die im Puffer abgelegt sind, d.h. der Mittelwert wird fast immer aus den LASTZKL-Werten der letzten 256 Zyklen gebildet.

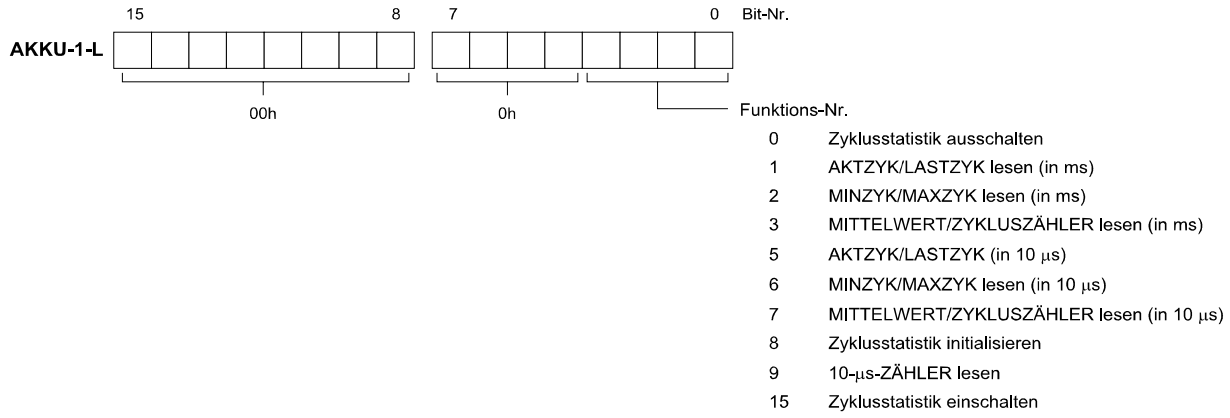
Funktionen

beim Aufruf des OB 152 können Sie durch Angabe einer Funktions-Nr. im AKKU-1-L verschiedene Einzelfunktionen aktivieren:

Parameter

AKKU-1-L

Der AKKU-1-L enthält die Funktions-Nr.; er muss folgenden Aufbau haben:



Fehlerfall

Der Fehlerfall tritt ein, wenn im AKKU-1-L eine falsche Funktions-Nr. übergeben wird (erlaubt sind nur die Nummern 0 bis 3, 5 bis 9 und 15).

Es wird der OB 31 (sonstige Laufzeitfehler) aufgerufen. Ist der OB 31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen wird in AKKU-1-L die Fehlerkennung 1A4Eh und in den AKKU-2-L die Fehlerkennung 0001h hinterlegt.

Ergebnis

Nach Aufruf des OB 152 sind die Anzeigen OS, OR und ERAB gelöscht (='0'), das VKE ist bis auf die nachfolgend aufgeführten Fälle ebenfalls ='0'. Außerdem werden in AKKU 1 und AKKU 2 die bei einigen Funktionen angeforderten Statistik-Größen übergeben (siehe nachfolgende Tabelle).

Funktionsergebnisse des OB 152

Funktion	Funktionsergebnisse		
	AKKU-1	AKKU-2	Bed. von "VKE =1"
Zyklusstatistik ausschalten	unverändert	unverändert	---
AKTZYK/ LASTZYK lesen	AKTZYK 3)	LASTZYK 3)	AKTZYK ist unkorrekt, Daten des akt. Zyklus werden nicht in die Statistik übernommen 1) oder Ergebniswert ist für Funktions-Nr zu groß (>FFFFh); als Ersatzwert wird FFFFh geliefert.
MINZYK/ MAXZYK lesen	MINZYK 3)	MAXZYK 3)	Ergebniswert ist für Funktions-Nr. 2 zu groß (>FFFFh); als Ersatzwert wird FFFFh geliefert.
MITTELWERT/ ZYKLUSZÄHLER lesen	MITTELWERT 3)	ZYKUSZÄHLER	Überlauf ZYKLUSZÄHLER 2) oder Ergebniswert ist für Funktions-Nr. 3 zu groß (>FFFFh); als Ersatzwert wird FFFFh geliefert.
Zyklusstatistik initialisieren	unverändert	unverändert	---
10-µs-ZÄHLER	10-µs-ZÄHLER	---	---
Zyklusstatistik einschalten	unverändert	unverändert	---

1) Durch WIEDERANLAUF

2) Ist beim Auslesen des Zykluszählers VKE ='1' gesetzt, so wird gleichzeitig mit der Anzeigenübergabe ein systeminterner Merker für Zyklusüberlauf gelöscht. Dieser Merker wird danach erst wieder gesetzt, wenn der Zykluszähler erneut übergelaufen ist.

3) Bei den Funktionen 1, 2 und 3 im Low-Word, bei den Funktionen 5, 6, 7 und 9 im High- und Low-Word der Akkus

Besonderheiten

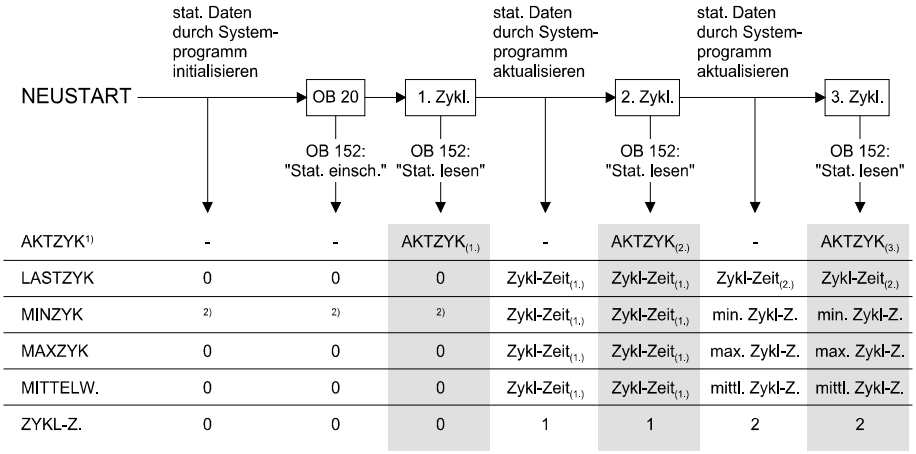
Nachfolgend erhalten Sie Hinweise auf einige Besonderheiten des OB 152 bei Neustart, nach einem ANLAUF oder bei Eintreffen bestimmter Ereignisse hin, die Sie bei seiner Benutzung beachten sollten.

Verhalten bei Neustart:

Bei NEUSTART werden die statistischen Daten initialisiert. Ein Aufruf des OB 152 im 1.Zyklus nach NEUSTART liefert die Initialisierungsdaten zurück.

Nachfolgendes Bild zeigt Ihnen, wie die statistischen Daten vom Systemprogramm nach Neustart und in den ersten drei Zyklen aktualisiert werden.

Verhalten nach Neustart



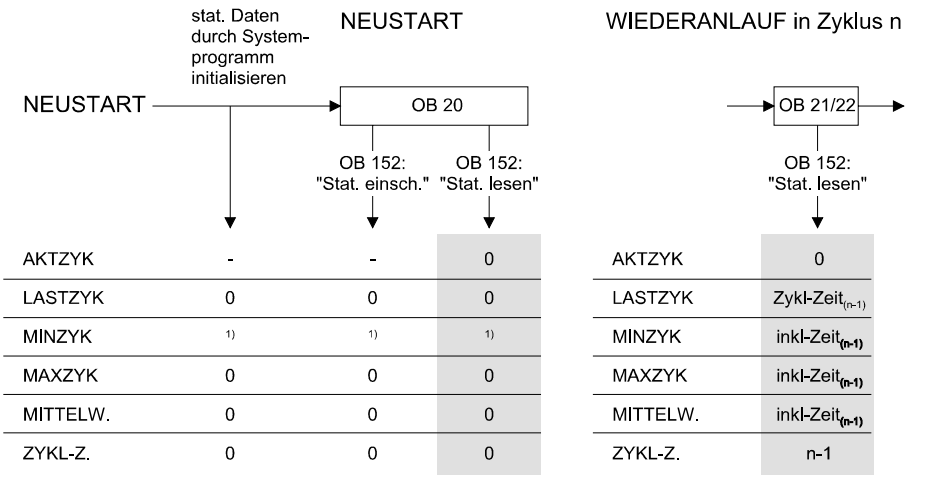
AKTZYK[1]	-	-	$AKTZYK_{(1.)}$	-	$AKTZYK_{(2.)}$	-	$AKTZYK_{(3.)}$
LASTZYK	0	0	0	$Zykl\text{-}Zeit_{(1.)}$	$Zykl\text{-}Zeit_{(1.)}$	$Zykl\text{-}Zeit_{(2.)}$	$Zykl\text{-}Zeit_{(2.)}$
MINZYK	2)	2)	2)	$Zykl\text{-}Zeit_{(1.)}$	$Zykl\text{-}Zeit_{(1.)}$	min. Zykl-Z.	min. Zykl-Z.
MAXZYK	0	0	0	$Zykl\text{-}Zeit_{(1.)}$	$Zykl\text{-}Zeit_{(1.)}$	max. Zykl-Z.	max. Zykl-Z.
MITTELW.	0	0	0	$Zykl\text{-}Zeit_{(1.)}$	$Zykl\text{-}Zeit_{(1.)}$	mittl. Zykl-Z.	mittl. Zykl-Z.
ZYKL-Z.	0	0	0	1	1	2	2

1) Der Wert für AKTZYK steht auch schon im 1. Zyklus zur Verfügung
2) Initialisierungswert (00FFFFh ms bzw. 0FF FFFFh * 10µs)

Beim Initialisieren der statistischen Daten werden, außer den in dem Bild aufgeführten Vorbesetzungen der Daten, systemintern der Umlaufpuffer für die Mittelwertbildung gelöscht sowie ein interner Merker für Zykluszählerüberlauf rückgesetzt.

Aufruf des OB 152 in einem Anlauf-OB:

Je nach Anlaufart liefert der Aufruf des OB 152 zum Lesen der statistischen Daten in AKKU-1-L und AKKU-2-L folgende Werte:

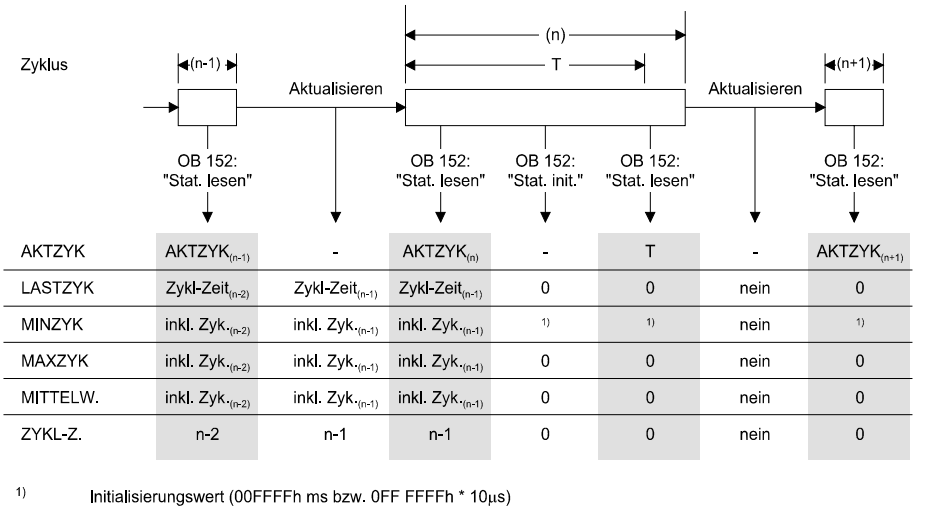


AKTZYK	-	-	0	AKTZYK	0
LASTZYK	0	0	0	LASTZYK	Zykl-Zeit$_{(n-1)}$
MINZYK	1)	1)	1)	MINZYK	inkl-Zeit$_{(n-1)}$
MAXZYK	0	0	0	MAXZYK	inkl-Zeit$_{(n-1)}$
MITTELW.	0	0	0	MITTELW.	inkl-Zeit$_{(n-1)}$
ZYKL-Z.	0	0	0	ZYKL-Z.	n-1

1) Initialisierungswert (00FFFFh ms bzw. 0FF FFFFh * 10µs)

Initialisierung der statistischen Daten durch Aufruf des OB 152:

Nachfolgendes Bild zeigt Ihnen, wie bei einer Initialisierung der statistischen Daten diese durch Aufruf des OB 152 im ZYKLUS verändert werden. Die grau unterlegten Spalten enthalten die beim Lesen der statistischen Daten übergebenen Werte.



AKTZYK	AKTZYK$_{(n-1)}$	-	AKTZYK$_{(n)}$	-	T	-	AKTZYK$_{(n+1)}$
LASTZYK	Zykl-Zeit$_{(n-2)}$	Zykl-Zeit$_{(n-1)}$	Zykl-Zeit$_{(n-1)}$	0	0	nein	0
MINZYK	inkl. Zyk.$_{(n-2)}$	inkl. Zyk.$_{(n-1)}$	inkl. Zyk.$_{(n-1)}$	1)	1)	nein	1)
MAXZYK	inkl. Zyk.$_{(n-2)}$	inkl. Zyk.$_{(n-1)}$	inkl. Zyk.$_{(n-1)}$	0	0	nein	0
MITTELW.	inkl. Zyk.$_{(n-2)}$	inkl. Zyk.$_{(n-1)}$	inkl. Zyk.$_{(n-1)}$	0	0	nein	0
ZYKL-Z.	n-2	n-1	n-1	0	0	nein	0

1) Initialisierungswert (00FFFFh ms bzw. 0FF FFFFh * 10µs)

Beim Initialisieren der statistischen Daten werden, außer den in der Tabelle aufgeführten Vorbesetzungen der Daten, systemintern der Umlaufpuffer für

die Mittelwertbildung gelöscht sowie ein interner Merker für Zykluszählerüberlauf rückgesetzt.

Nach dem Initialisieren der statistischen Daten durch Aufruf des OB 152 werden die Daten durch das Systemprogramm erst am Ende des 1. Zyklusses, der nach der Initialisierung folgt, aktualisiert.

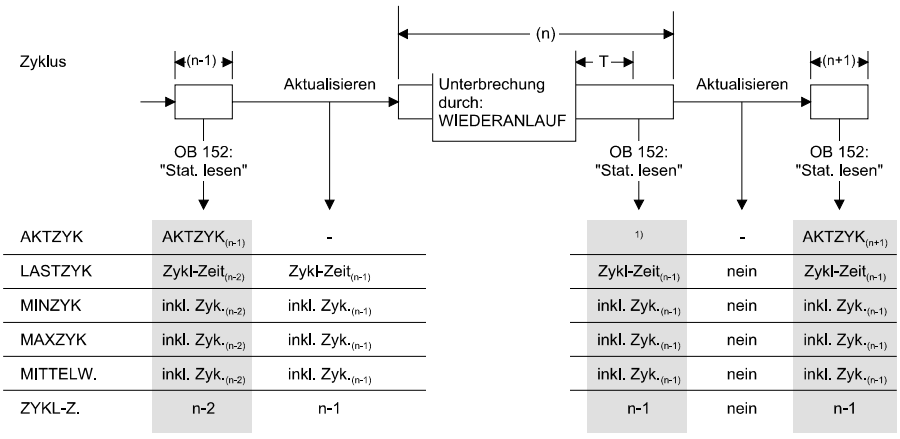
Aufruf des OB 152 bei ausgeschalteter Zyklusstatistik:

Wird die Zyklusstatistik durch Aufruf des OB 152 ausgeschaltet, so bleiben die statistischen Daten der letzten Aktualisierung erhalten. Wird danach der OB 152 zum Lesen von Statistik-Daten aufgerufen, so liefert er entsprechend die Daten der letzten Aktualisierung vor dem Ausschalten.

Werden die statistischen Daten nach einem NEUSTART gelesen, ohne dass die Zyklusstatistik durch Aufruf des OB 152 eingeschaltet wurde, so liefert der OB 152 die Initialisierungsdaten.

Verfälschung der statistischen Daten

Durch das Auftreten eines WIEDERANLAUFS wird die Erfassung der Zyklusdauer für den aktuellen Zyklus gestört und führt dann zu falschen Werten. Daher werden in diesem Fall die statistischen Daten für den betroffenen Zyklus nicht aktualisiert.



AKTZYK	$AKTZYK_{(n-1)}$	-	1)	-	$AKTZYK_{(n+1)}$
LASTZYK	$Zykl\text{-}Zeit_{(n-2)}$	$Zykl\text{-}Zeit_{(n-1)}$	$Zykl\text{-}Zeit_{(n-1)}$	nein	$Zykl\text{-}Zeit_{(n-1)}$
MINZYK	$inkl.\ Zyk._{(n-2)}$	$inkl.\ Zyk._{(n-1)}$	$inkl.\ Zyk._{(n-1)}$	nein	$inkl.\ Zyk._{(n-1)}$
MAXZYK	$inkl.\ Zyk._{(n-2)}$	$inkl.\ Zyk._{(n-1)}$	$inkl.\ Zyk._{(n-1)}$	nein	$inkl.\ Zyk._{(n-1)}$
MITTELW.	$inkl.\ Zyk._{(n-2)}$	$inkl.\ Zyk._{(n-1)}$	$inkl.\ Zyk._{(n-1)}$	nein	$inkl.\ Zyk._{(n-1)}$
ZYKL-Z.	n-2	n-1	n-1	nein	n-1

1) Der Wert für AKTZYK entspricht der Zeit T, die seit dem Auftreten der Störung im aktuellen Zyklus verstrichen ist. Dies ist nicht die Dauer des gesamten Zyklus. Um diesen Fall zu kennzeichnen, wird zusätzlich zu den in AKKU-1-L und AKKU-2-L übergebenen Werten das VKE='1' gesetzt.

OB 160 bis 163: Zählschleifen

Einleitung

Mit Hilfe dieser Sonderfunktions-Organisationsbausteine realisieren Sie Programmschleifen mit besonders günstiger Laufzeit.

Funktion

Die vier Organisationbausteine OB 160, OB 161, OB 162 und OB 163 ermöglichen eine vierfache Schachtelung von Schleifen. Sie können damit in den Systemdatenwörtern BS 60 bis 63 vier verschiedene Schleifenzähler einsetzen.

Jedem der vier Obs ist ein bestimmtes Systemdatenwort zugeordnet:

- BS 60: OB 160
- BS 61: OB 161
- BS 62: OB 162
- BS 63: OB 163.

Programmieren der Programmschleife

In eines der Systemdatenwörter transferieren Sie den Wert für die erwünschte Anzahl an Schleifendurchläufen. Rufen Sie nun den dazugehörigen Sonderfunktions-OB auf, so wird der Schleifenzähler im Systemdatenwort um eins erniedrigt. Die Schleife wird solange durchlaufen, bis der Schleifenzähler den Wert Null enthält.

Hinweis!

Enthält der Schleifenzähler bereits vor Aufruf des Sonderfunktions-OBs den Wert Null, so wird er bei Aufruf ebenfalls um eins erniedrigt: Es erfolgen 65536 Schleifendurchläufe!

Parameter

Systemdatenwörter BS 60 bis 63

Schleifenzähler,

zulässige Werte: 0 bis 65 535 dezimal (0 bis FFFFh).

Ergebnis

Schleifenzähler im Systemdatenwort >0: VKE wird gesetzt (VKE =1)

Schleifenzähler im Systemdatenwort =0: VKE wird gelöscht (VKE =0)

Die restlichen Bit- und Wortanzeigen werden immer gelöscht.

Die AKKUs werden nicht verändert und nicht ausgewertet. Somit stehen diese zu Beginn des nächsten Schleifendurchlaufs noch zur Verfügung und müssen nicht neu geladen werden.

Fehlerfälle

keine

Beispiel Zählschleife programmieren:

```
Im Merkerwort x steht die gewünschte Anzahl der
Schleifendurchläufe.
          :
          :              Schleife initialisieren
          :L   KB 0
          :L   MW x      Schleifenzähler
          :!=  F
          :SPB =M002
          :T   BS 62     Schleifenzähler ins
          :              Systemdatenwort transferieren
          :
M001      :              "Schleifenprogramm":
          :       .
          :       .
          :       .
          :              Schleife verwalten:
          :SPA OB 162    Zählschleife
          :SPB =M001
          :
          :
M002      :              weiter
          :       .
          :       .
          :       .
```

OB 170: Bausteinstack (BSTACK) lesen

Einleitung

Im Bausteinstack sind, ausgehend vom OB 1 bzw. FB 0, alle Bausteine eingetragen, die nacheinander aufgerufen worden sind und deren Bearbeitung noch nicht abgeschlossen ist.

Funktion

Mit Hilfe des Sonderfunktions-Organisationsbausteins OB 170 können Sie die im BSTACK vorhandenen Einträge in einen Datenbaustein einlesen. Auf diese Weise ermitteln Sie die vorhandene Anzahl an BSTACK-Einträgen und damit die Reserve, die Ihnen für weitere Einträge noch zur Verfügung steht.

Zu jedem Eintrag erhalten Sie die jeweilige Rücksprungadresse (STEP-Adresszähler =SAZ), die absolute Anfangsadresse des in diesem Baustein gültigen Datenbausteins (DBA) sowie dessen Länge (Anzahl der Datenwörter =DBL).

Hinweis

Vor Aufruf des OB 170 muss ein ausreichend langer Datenbaustein (DB oder DX) aufgeschlagen werden. Für jeden gewünschten BSTACK-Eintrag benötigen Sie vier Datenwörter.

Parameter

Akkus

AKKU-2-L

Nummer des Datenwortes (DWn), ab dem die Einträge im aufgeschlagenen DB abgelegt werden sollen (Offset).

AKKU-1-L

gewünschte Anzahl an BSTACK-Elementen,

zulässige Werte: 1 bis 62.

Beispiel:

Enthält der AKKU-1-L den Wert '1', so erhalten Sie den letzten BSTACK-Eintrag,

bei '2' den letzten und vorletzten, usw..

Ergebnis

Nach erfolgreichem Aufruf des OB 170:

- steht im AKKU-2-L weiterhin der Offset im Datenbaustein,
- steht im AKKU-1-L die tatsächlich dargestellte Anzahl an BSTACK-Elementen [1)]
- wird das VKE gelöscht,
- können die Ergebnisanzeigen ANZ 0 und ANZ 1 ausgewertet werden,
- sind alle übrigen Bit- und Wortanzeigen gelöscht.

[1)] mögliche Werte: 0 bis 62, wobei dargestellte Anzahl ≤ gewünschte Anzahl, 0 = "kein BSTACK-Eintrag vorhanden" oder "Fehler" (Inhalt von AKKU-1-L muliplziert mit 4 ergibt die Anzahl der beschriebenen Datenwörter im aufgerufenen DB).

Beeinflussung der Ergebnisanzeigen VKE, ANZ 1 und ANZ 0

VKE	ANZ 1	ANZ 0	Abfrage mit	Bedeutung
0	0	1	SPM	vorhandene Anz. BSTACK-Elemente. <gewünschte Anz.
0	0	0	SPZ	vorhandene Anz BSTACK-Elemente =gewünschte Anz.
0	1	0	SPP	vorhandene Anz BSTACK-Elemente >gewünschte Anz.
1	1	1	SPB	Fehler

Ablage der BSTACK-Elemente im aufgeschlagenen Datenbaustein

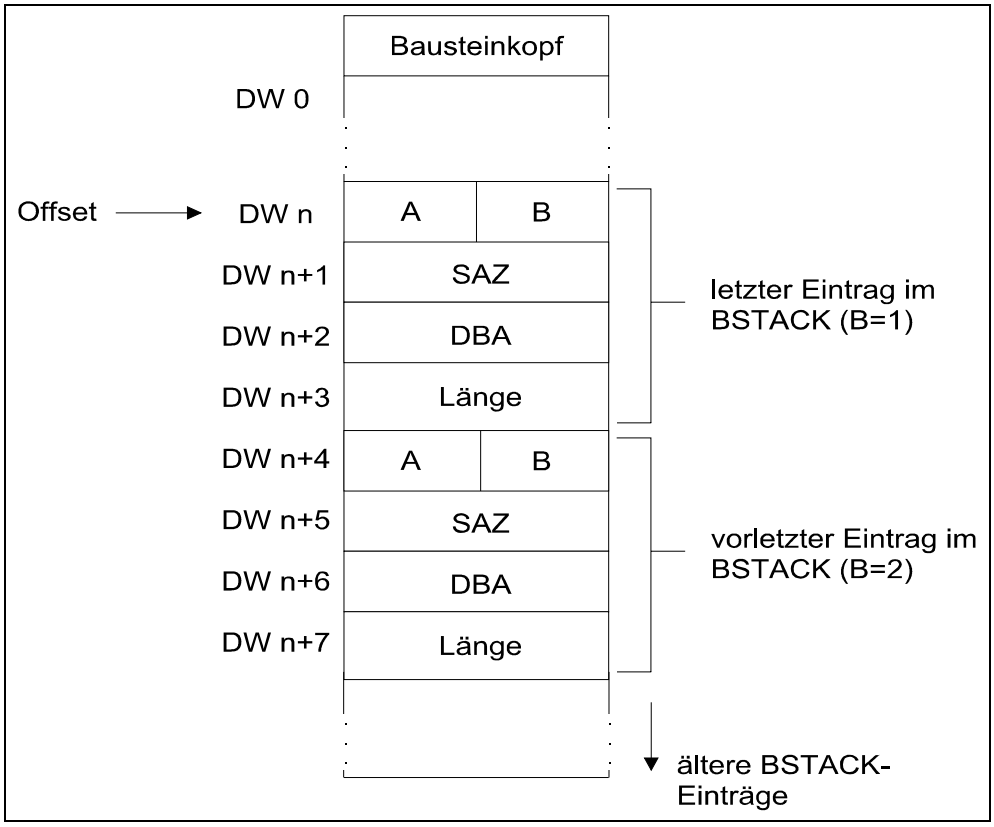
So werden die Inhalte des BSTACKs bei Aufruf des OB 170 im aufgerufenen Datenbaustein abgelegt:

A = BSTACK-Element-Nummer (62 bis 1)

(Bereits bei Ausgabe des letzten BSTACK-Elements lässt sich so die Reserve ermitteln:

A = 17 Reserve = A-1 = 16)

B = Tiefe des BSTACK-Elementes (1 bis 62)



Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- kein Datenbaustein aufgeschlagen,
- aufgeschlagener Datenbaustein nicht vorhanden oder nicht ausreichend lang , um die gewünschte Anzahl von BSTACK-Einträgen aufnehmen zu können,
- unzulässige Parameter in AKKU 1 und AKKU 2.

Im Fehlerfall werden das VKE, sowie die Ergebnisanzeigen ANZ 0 und ANZ 1 gesetzt (VKE, ANZ 0 und ANZ 1 =1), die übrigen Bit- und Wortanzeigen werden gelöscht.

Der Inhalt von AKKU-1-L wird zu '0'.

Beispiel

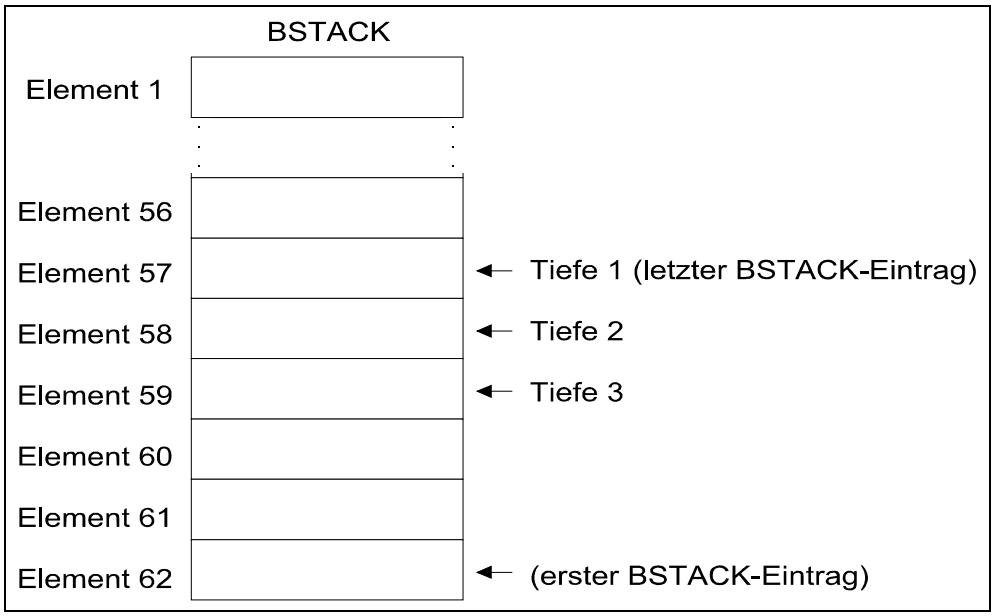
Sie wollen die letzten 3 BSTACK-Einträge in den Datenbaustein DX 10 einlesen.

Die Einträge sollen in DX 10 ab Datenwort 16 abgelegt werden (Siehe Bilder KEIN MERKER).

```
:AX     DX 10       Aufschlagen DX 10
:L      KY 0,16     BSTACK-Einträge sollen ab DW 16
:                   abgelegt werden
:L      KY 0,3      gewünscht werden die letzten 3
:                   BSTACK-Einträge
:SPA    OB170
```

Im BSTACK sind 6 Bausteine eingetragen:

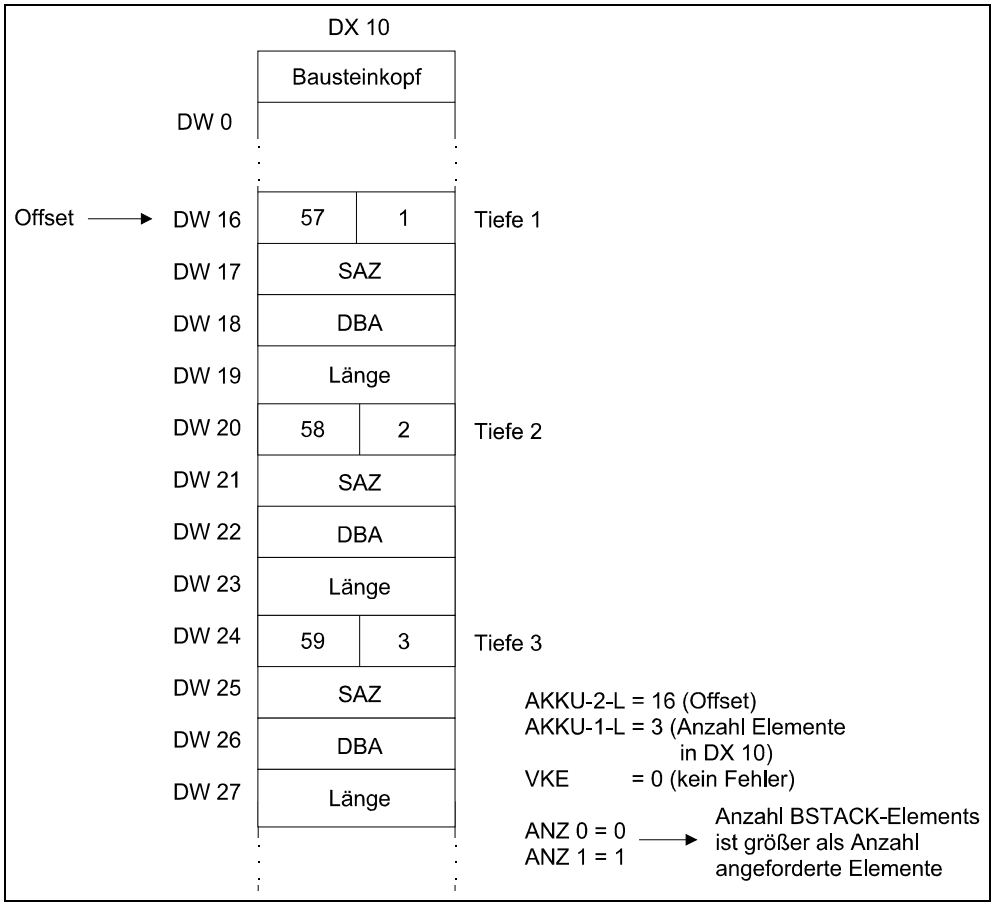
BSTACK-Belegung im Beispiel



Fortsetzung des Beispiels:

Nach Aufruf des OB 170 ist der Datenbaustein DX 10 folgendermaßen belegt.

Belegung DX10 nach Aufruf des OB 170



OB 180: Variabler Datenbaustein-Zugriff

Funktion

Mit dem OB 180 wird die Anfangsadresse des aktuellen Datenbausteins um einen vorgegebenen Wert verschoben. Dabei wird berücksichtigt, dass die noch verfügbare Länge des DBs abnimmt (DBA- und DBL-Register werden entsprechend der Verschiebung geladen).

Hinweis!

Vor Aufruf des OB 180 muss ein ausreichend langer Datenbaustein (DB oder DX) aufgeschlagen sein.

DBA-DBL-Register

Beim Aufschlagen eines Datenbausteins mit den Operationen A DB und AX DX wird das "DBA"-Register (Datenbaustein-Anfangsadresse) mit der Adresse des Datenwortes DW 0 geladen, die im DB 0 hinterlegt ist.

Zugriffe auf Datenbausteine mit Operationen wie L DR 60 oder B DW 240 usw. erfolgen immer relativ zur Datenbaustein-Anfangsadresse.

Zusätzlich zum DBA-Register wird bei jedem Aufruf eines Datenbausteins das "DBL"-Register (Datenbaustein-Länge)geladen: Es enthält die Länge (in Worten) des aufgeschlagenen DB- oder DX-Datenbausteins ohne Baustein-Kopf.

Hinweis!

Im DBL-Register kann eine maximale Länge von bis zu 4091 Datenwörtern eingetragen sein.

Zugriffe auf Datenwörter können nur auf Datenwort-Nummern bis 255 erfolgen.

Beispiel

Das DBA-Register enthält die Adresse des Speicher-Wortes, in welchem das DW 0 des DB 17 hinterlegt ist: DBA = 151Bh.

Im DBL-Register ist die Anzahl der Datenwörter hinterlegt:

DBL = 8 (DW 0 bis DW 7).

Da der Zugriff auf Datenwörter mittels der Operationen L DW, U D, B DW usw. immer relativ zum DBA erfolgt, wird - um z.B. auf das Wort DW 3 zuzugreifen - 3 zu 151Bh addiert. Unter der Adresse 151Eh ist das Datenwort DW 3 abgelegt.

Anhand des DBL-Registers wird geprüft, ob ein Transfer- oder Ladefehler vorliegt. So ist z.B. T DW 7 erlaubt, T DW 8 bzw. L DW 8 jedoch fehlerhaft.

Anwendung des OB 180

Der Sonderfunktions-OB 180 bietet Ihnen die Möglichkeit, auf strukturierte Daten in einem aufgeschlagenen Datenbaustein zuzugreifen. Dies können Sie dadurch erreichen, dass Sie die im Register DBA eingetragene Anfangsadresse des Datenbausteins mit Hilfe des OB 180 zum Ende des Datenbausteins hin verschieben. Gleichzeitig mit einer Verschiebung der Anfangsadresse wird durch OB 180 die im Register DBL eingetragene Bausteinlänge entsprechend vermindert. Dies ist wichtig, damit die CPU bei später erfolgenden Zugriffen auf den Datenbaustein eine Überwachung bei Lade- und Transferoperationen durchführen kann.

- Arbeiten mit Datenbausteinen, die eine Länge größer 261 Wörter (5 Wörter Kopf) haben: Mit Hilfe des OB 180 können Sie ein "Zugriffsfenster" von 256 Datenwörtern beliebig über einem Datenbaustein verschieben.
- Anwendung bei Datenstrukturen:
  Ein Datenbaustein kann in mehrere Datensätze mit gleicher Länge und gleicher Anordnung der darin enthaltenen Daten unterteilt sein. Man spricht in diesem Fall von einer Strukturierung des Datenbausteins.
  Ein so strukturierter Datenbaustein kann z. B. die Daten zu mehreren Teilprozessen enthalten, wobei im ersten Datenwort ein Temperaturwert, im zweiten ein Druck und in den übrigen Datenwörtern andere Messwerte des Teilprozesses abgelegt sind.
  Mit Hilfe des OB 180 können Sie auf die Daten jedes dieser Teilprozesse mit denselben Operationen (z. B.L DD, S D, T DR usw.) zugreifen, indem Sie das DBA-Register jeweils mit der Anfangsadresse für die Teilprozessdaten laden.
  Im Gegensatz zu anderen Substitutionsmechanismen (Substitution = indizierte Parametrierung) ergeben sich bei diesem Verfahren einfachere und laufzeitgünstigere Programme.

Parameter

AKKU-1-L

Versatz, Anzahl der Datenwörter, um die die DB-Anfangsadresse verschoben werden soll

zulässige Werte: 0<=AKKU-1-L<DBL

Ergebnis

Nach erfolgreichem Aufruf des OB 180:

- ist der Wert des DBA-Registers (=Adresse des DW 0) um den Wert des AKKU-1-L erhöht,
- ist der Wert des DBL-Registers um den Wert des AKKU-1-L erniedrigt,
- ist das VKE gelöscht (VKE = 0),
- sind alle übrigen Bit- und Wortanzeigen gelöscht.

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- negative Länge,
- kein Datenbaustein aufgeschlagen,
- Inhalt AKKU-1-L ≥ DBL.

Im Fehlerfall (Inhalt AKKU-1-L ≥ DBL) bleiben DBA- und DBL-Register unbeeinflusst. Das VKE wird gesetzt (VKE =1). Die übrigen Bit- und Wortanzeigen werden gelöscht.

Enthält das DBL-Register den Wert '0', so erkennt der OB 180, dass kein Datenbaustein aufgeschlagen ist. Das VKE wird gesetzt (VKE =1) und signalisiert so einen Fehler.

DBA und DBL auf Anfangswert zurückstellen

Ein erneutes Aufschlagen des Datenbausteins mit den Operationen A DB oder AX DX stellt den Grundzustand wieder her.

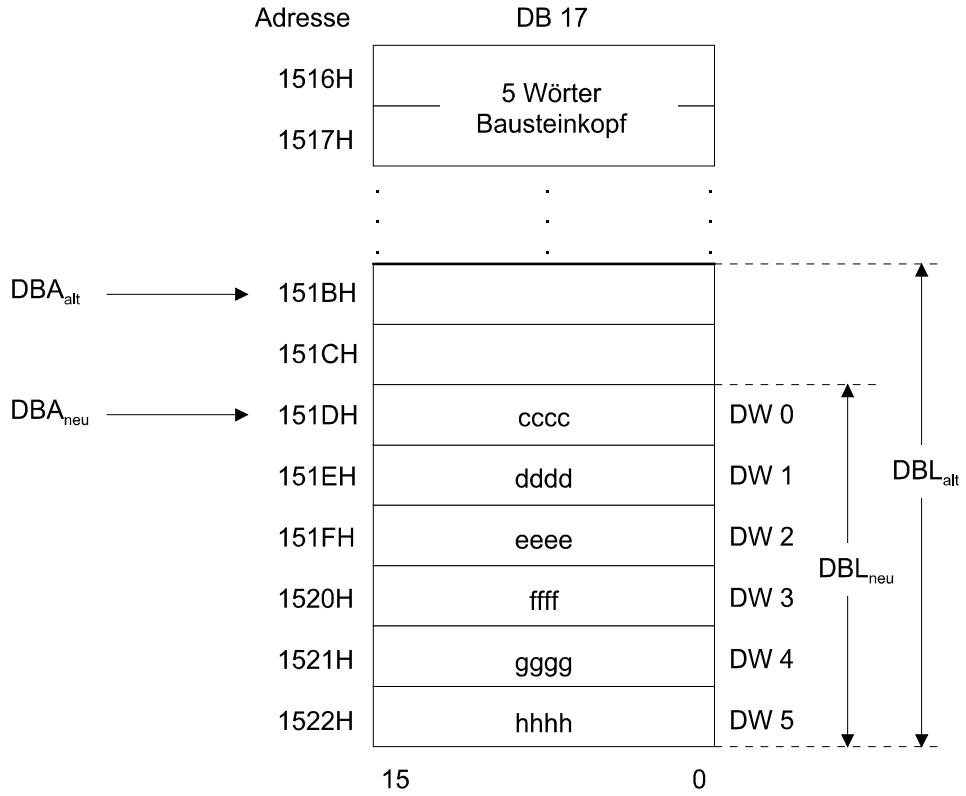
Beispiel

Die Datenbausteinanfangsadresse (DBA = 151Bh) im DB 17 (DBL = 8) soll um zwei Datenwörter verschoben werden.

```
:A    DB 17     DB 17 aufschlagen
:L    KB 2      Verschiebung/Versatz als Konstante
:SPA  OB 180    OB 180 aufrufen: DBA und DBL werden
                angepasst.
```

Nach Aufruf des OB 180 lässt sich z.B. unter der Adresse 1520h gespeicherte Datenwort nicht mehr mit DW 5, sondern mit DW 3 ansprechen usw.

Fortsetzung des Beispiels



Aufgrund des gleichzeitigen veränderten DBL-Registers bleibt die Fehler-Überwachung gewährleistet: Die Operation T DW 5 ist erlaubt, während T DW 6 bzw. L DW 6 fehlerhaft sind

Durch erneute Aufrufe des OB 180 kann das DBA weiter erhöht werden (wobei das DBL-Register weiter verringert wird): Die Operation A DB17 stellt den Grundzustand (DBA =151Bh, DBL = 8) wieder her.

Hätte der DB 17 eine Länge von z. B 258 Datenwörtern, so könnten Sie mit Operationen nicht mehr auf DW 256 und DW 257 zugreifen. Durch Verschieben des DBA-Registers um 2 lassen sich die Datenwörter 256 und 257 mit "DW 254" und "DW255" ansprechen.

(Zum DBA-/DBL-Register siehe auch Band 1 / Teil 9).

OB 181: Datenbausteine (DB/DX) testen

Einleitung

Mit dem Sonderfunktions-Organisationsbaustein OB 181 können Sie prüfen,

- ob ein bestimmter DB- oder DX-Datenbaustein vorhanden ist,
- unter welcher Adresse das erste Datenwort des Datenbausteins abgelegt ist,
- wieviele Datenwörter dieser Datenbaustein enthält,
- welcher Speichertyp und Speicherbereich benutzt wird.

Anwendung des OB 181

Eine Anwendung der Funktion "DB/DX testen" ist sinnvoll vor den Befehlen TNB/TNW, E DB/EX DX und vor Aufruf der Sonderfunktions-Organisationsbausteine OB 182, OB 254 und OB 255.

So können Sie z. B. vor einem Blocktransfer von Datenwörtern den OB 181 aufrufen, um sicherzustellen, dass alle zu kopierenden Datenwörter aufzunehmen sind.

Funktion

Der OB 181 prüft, ob ein Datenbaustein vorhanden ist, und gibt als Ergebnis die charakteristischen Parameter des Datenbausteins zurück.

Parameter

AKKU-1-L

AKKU-1-LL

Bausteinnummer,

zulässige Werte: 1 bis 255

AKKU-1-LH

Bausteinkennung,

zulässige Werte: 1=DB

2=DX

Ergebnis

- Der geprüfte Baustein ist in der CPU vorhanden:

  AKKU-1-L: Adresse des 1. Datenwortes (DW0),

  AKKU-2-L: Länge des Datenbausteins in Worten (ohne Bausteinkopf)
  Beispiel: In AKKU-2-L steht der Wert '7':
  Der Datenbaustein besteht aus den Datenwörtern DW 0 bis DW 6,

  VKE: = 0,

  ANZ 1/ANZ 0: werden entsprechend der Lage des Bausteins beeinflusst (siehe nachfolgende Tabelle),

  restliche Bit-und Wortanzeigen: werden gelöscht.

- Der geprüfte Baustein ist in der CPU nicht vorhanden oder es ist ein Fehler aufgetreten:

  AKKU 1 und 2: werden nicht verändert,

  VKE: = 1,

  ANZ 1/ANZ 0: = 1,

  restliche Bit-und Wortanzeigen: werden gelöscht.

VKE, ANZ 1, ANZ 0 Entsprechend dem Prüfergebnis werden folgende Anzeigen gesetzt, die durch die in der Tabellenspalte "Abfrage" aufgeführten Operationen ausgewertet werden können:

VKE	ANZ 1	ANZ 0	Abfrage	Bedeutung		
0	0	1	SPM	DB/DX im Anwenderspeicher	DB/DX im EPROM-Betrieb (nur lesbar)	DB/DX vorhanden
0	0	0	SPZ	DB/DX im Anwenderspeicher	DB/DX im RAM-Betrieb (schreib- und lesb.)	DB/DX vorhanden
0	1	0	SPP	DB/DX im DB-RAM	DB/DX im RAM-Betrieb (schreib- und lesb.)	DB/DX vorhanden
		1	1	DB/DX nicht vorhanden oder Fehler		

Fehlerfälle Es werden folgende Fehlerfälle unterschieden

- falsche Bausteinnummer (unzulässig: 0 -DB 0 oder DX 0),
- falsche Bausteinkennung (zulässig: 1 = DB, 2 = DX; unzulässig: 0, 3 bis 255),
- Speicherfehler.

OB 182: Datenbereich kopieren

Funktion

Der OB 182 kopiert einen Datenblock variabler Länge von einem Datenbaustein in einen anderen. Als Quell- und Zielbausteine können DB- und DX-Datenbausteine verwendet werden. Der Blockanfang im Quell- und Zieldatenbaustein ist frei wählbar. Der OB 182 kann maximal 4091 Datenwörter kopieren. Er enthält Pseudobefehlsgrenzen.

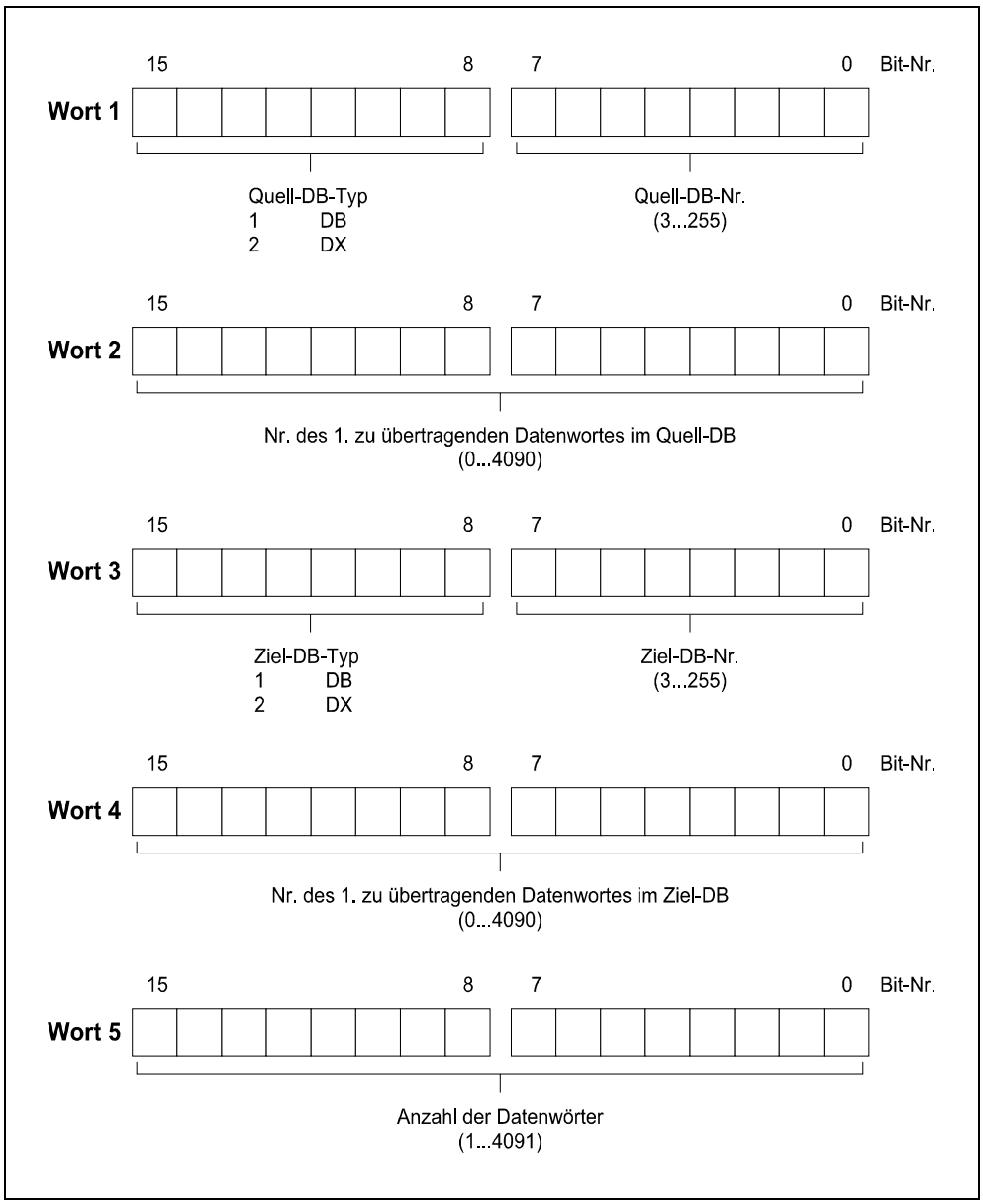
Hinweis!

Quell- und Zielbaustein können identisch sein. Die Datenbereiche von Quelle und Ziel dürfen sich überlappen. Die Originaldaten des Quellbereichs werden auch bei Überlappung unverändert in den Zielbereich kopiert. Der Überlappungsbereich in der Quelle ist nach dem Kopiervorgang überschrieben. Mit dieser Funktionseigenschaft können Sie einen Datenbereich innerhalb eines Bausteins verschieben.

Parameter

1. Datenfeld Kopierparameter

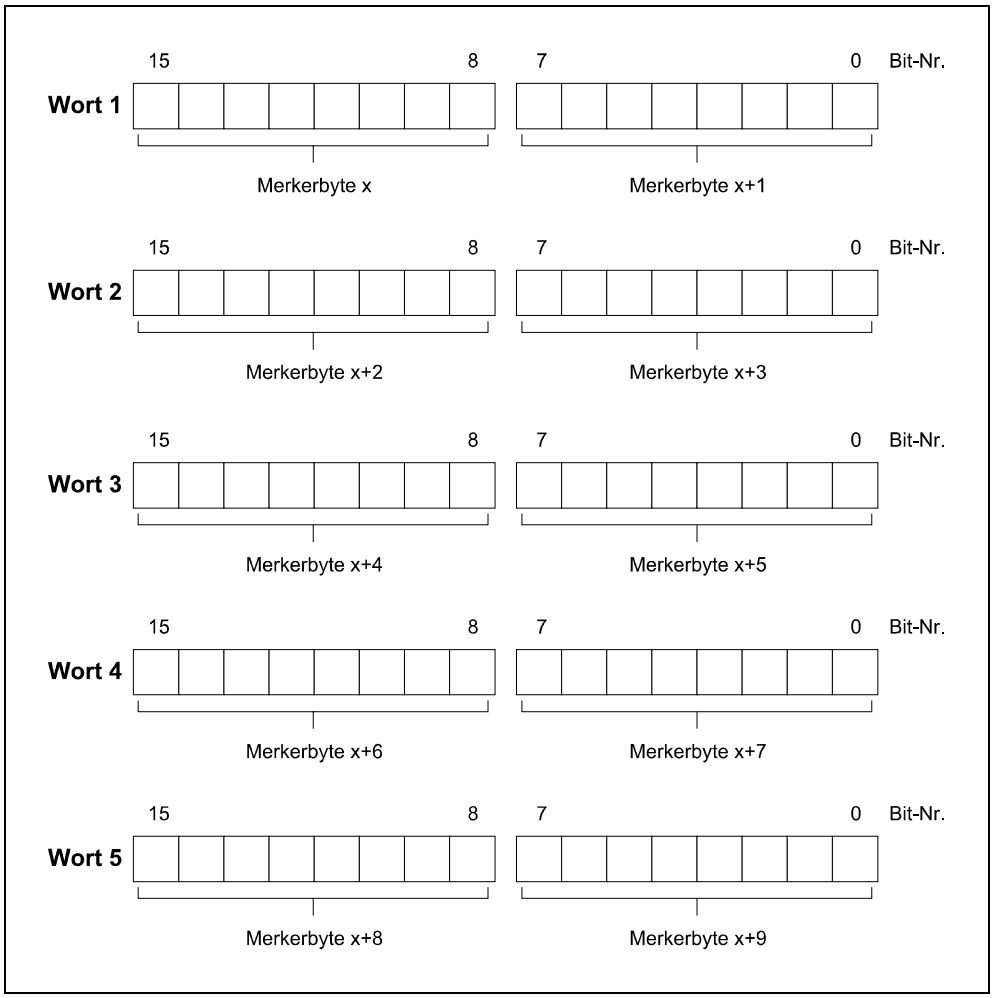
Vor dem Aufruf des OB 182 versorgen Sie ein Datenfeld mit den erforderlichen Parametern für den gewünschten Kopiervorgang. Dieses Datenfeld kann in einem DB- oder DX-Datenbaustein, im M- oder S-Merkerbereich angelegt werden. Das Datenfeld (5 Worte) bezeichnet Quell- und Zieldatenbaustein, die Blockanfangsadresse in beiden Bausteinen sowie die Anzahl der zu übertragenden Datenwörter.



Datenfeld im Merkerbereich:

Wenn Sie das Datenfeld in einem Merkerbereich anlegen, müssen Sie folgende Zuordnung der Datenfeldwörter zu den Merkerbytes berücksichtigen. Dabei ist 'x' der Parameter "Nr. des 1.Datenfeldwortes", den Sie beim Aufruf des OB 182 im AKKU-1-L hinterlegen müssen:

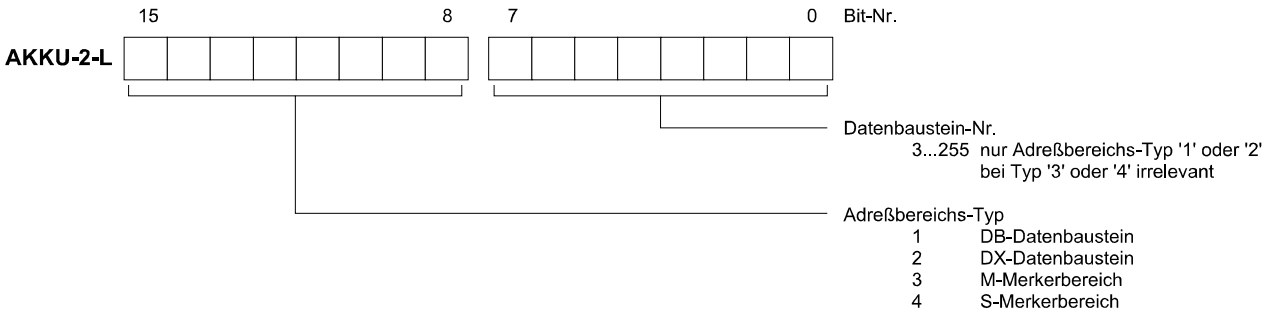
Datenfeld im Merkerbereich



2. AKKUs

AKKU-2-L

Der AKKU-2-L enthält Angaben zum verwendeten Datenfeld. Er muss folgenden Aufbau haben:



AKKU-1-L

Nummer des 1.Datenfeldwortes,

zulässige Werte (in Abhängigkeit vom Adressbereichs-Typ):

DB, DX:	0 bis 2043
M-Merker:	0 bis 246 (= Nr. Merkerbyte 'x')
S-Merker:	0 bis 1014 (= Nr. Merkerbyte 'x')

Ergebnis

Nach korrekter Bearbeitung des OB 182 sind die Anzeigenbits OR, ERAB und OS =0. Alle anderen Anzeigenbits sowie AKKU 1 und AKKU 2 bleiben unverändert.

Fehlerfälle

Im Fehlerfall werden der OB 19 oder der OB 31 aufgerufen. Sind der OB 19 oder der OB 31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen werden Fehlerkennungen in den AKKU 1 und AKKU 2 hinterlegt (siehe nachfolgende Tabelle)

Fehlerkennungen des OB 182

AKKU-1-L	AKKU-2-L	Bedeutung	aufger. OB
1A06h	---		OB 19
1A34h	0001h	Beschreibung des Datenfeldes fehlerhaft	OB 31
	0100h	Adressbereichs-Typ unzulässig	
	0101h	DB-Nr. unzulässig	
	0102h	"Nr. des ersten Datenfeldwortes" unzulässig	
	0200h	"Quell-DB-Typ" unzulässig	
	0201h	"Quell-DB-Nr." unzulässig	
	0202h	"Nr. des 1. zu übertr. Datenwortes im Quell-DB" unzulässig	
	0203h	Länge Quelldatenbaustein im Bausteinkopf <Wörter	
	0210h	"Ziel-DB-Typ" unzulässig	
	0211h	"Ziel-DB-Nr." unzulässig	
	0212h	"Nr. des 1. zu beschreibenden DW im Ziel-DB" unzulässig.	
	0213h	Länge Ziel_DB im Bausteinkopf < 5 Wörter	
	0220h	"Anzahl zu übertrag. DW" unzulässig (= 0 oder > 4091)	
	0221h	Quell-DB zu kurz	
	0222h	Ziel-DB zu kurz	
	0223h	Ziel-DB schreibgeschützt (EPROM-Betrieb)	

OB 186: Speicher komprimieren

Funktion

Mit dem Organisationsbaustein OB 186 können Sie den Speicher komprimieren und den Status der Komprimierfunktion abfragen bzw. prüfen. Die Funktionalität des OB 186 entspricht der PG-Funktion "Speicher komprimieren".

Hinweis!

Während der Speicher mittels des OB 186 komprimiert wird, wird Die PG-Funktion "Speicher komprimieren" abgewiesen. Andere PG-Funktionen sind nur eingeschränkt zulässig.

Während PG-Funktionen aktiv sind, wird der OB 186 abgewiesen.

"Speicher komprimieren" mit dem OB 186 ist eine langlaufende Funktion, die vom Systemprogramm auf mehrere Zyklen verteilt wird. Es werden keine expliziten Rückmeldungen ausgegeben. Rückmeldungen können jedoch durch zyklisches Aufrufen des OB mit Funktionsnummer 2 realisiert werden.

Der OB 186 ruft keinen Fehler-OB auf.

Parameter

AKKU-1-L

Der AKKU-1-L enthält die Funktionsnummer,

zulässige Werte: 0001h: Komprimieren starten

0001h: Status der Komprimierfunktion abfragen / prüfen.

Ergebnis

Der OB 186 setzt in den Bitanzeigen OS und /ERAB sowie in den Wortanzeigen OR auf 0.

Der Aufruf des OB 186 bringt folgende Ergebnisse:

Funkt.-Nr	VKE	AKKU-1-L	Bedeutung
0001h	1	0000h	Kein Fehler, Kompr. wurde angestossen
0001h	0	8002h	Fehler, Starten nicht möglich, da PG-Funkt. aktiv
0001h	0	8003h	Fehler, Starten nicht möglich, da Kompr. bereits aktiv war
0002h	1	0000h	Kein Fehler, Kompr. läuft noch
0002h	0	0001h	Kompr. beim letzten Aufruf war nicht mögl, da CPU zu diesem Zeitpunkt nicht in komprimierfähigen Zustand war
0002h	0	8002h	Komprimieren abgeschl. ohne Fehler
0002h	0	0003h	Komprimieren abgebrochen wegen Speicherinkonsistenz
ungültig	0	8001h	Fehler, ungültige Funktionsnummer

OB 190/192: Merker in Datenbaustein übertragen

Anwendung

Die Organisationsbausteine OB 190 und OB 192 übertragen eine vom Anwender vorgegebene Anzahl Merkerbytes in einen dafür vorgesehenen Datenbaustein.

Dies kann z. B. von Vorteil sein vor Bausteinaufrufen, im Fehler-Organisationsbaustein oder bei Unterbrechung durch eine zeit- oder alarmgesteuerte Programmbearbeitung.

Mit Hilfe der Organisationsbausteine OB 191 und OB 193 können Sie diese Merkerbytes anschließend wieder aus dem Datenbaustein zurückschreiben.

Hinweis!

Verwenden Sie OB 190 und OB 191 für das einfache Retten und Zurücklesen der Merkerbytes, da Sie damit erhebliche Laufzeitvorteile gewinnen.

Vor dem Aufruf von OB 190/192 muss ein Datenbaustein (DB/DX) aufgeschlagen werden.

Die OB 190/192 übertragen **nur aus dem M-Merkerbereich**, jedoch nicht aus dem **S-Merkerbereich** in einen Datenbaustein.

Funktion

Nach Aufruf des OB 190/192 werden im aufgeschlagenen DB die Merkerbytes ab der angegebenen Datenwortadresse gespeichert. Den Bereich der zu rettenden Merker entnehmen OB 190/192 dem AKKU 2.

OB 190 und OB 192 sind identisch mit Ausnahme der Art und Weise, in der sie die Merkerbytes übertragen:

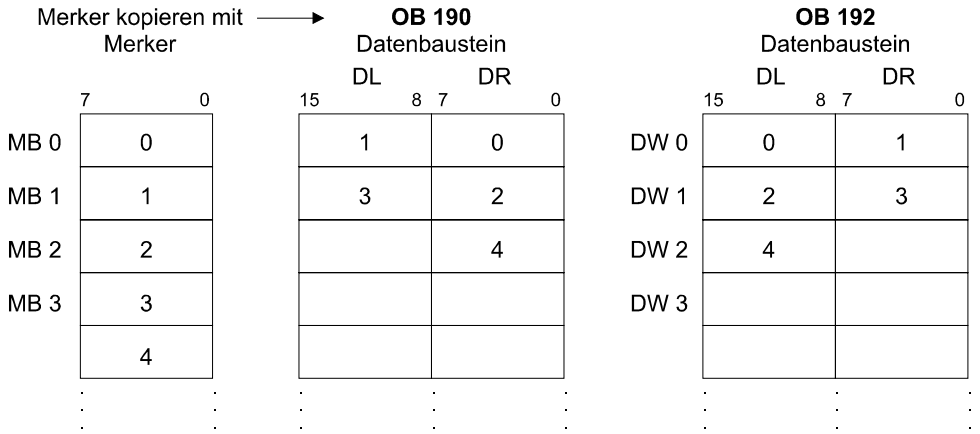
OB 190 überträgt die Merker **byteweise,**

OB 192 überträgt die Merker **wortweise**.

Dies ist von Belang, wenn die in den Datenbausteinen übertragenen Daten anschließend bearbeitet werden sollen und der Datenbaustein nicht nur als einfacher Zwischenspeicher benutzt wird.

Die folgende Abbildung soll diesen Unterschied verdeutlichen:

Byteweises und wortweises Übertragen



Hinweis

Falls eine ungerade Anzahl von Merkerbytes übertragen wird, so wird das letzte benutzte Datenwort des Datenbausteins nur zur Hälfte genutzt. Bei OB 190 bleibt das Datum links, bei OB 192 das Datum rechts im Ziel-DB unverändert.

Parameter

1. Angaben zur Datenquelle:

AKKU-2-LH

Erstes zu übertragendes Merkerbyte,

zulässige Werte: 0 bis 255.

AKKU-2-LL

Letztes zu übertragendes Merkerbyte,

zulässige Werte: 0 bis 255,

(Letztes Merkerbyte ≥Erstes Merkerbyte).

2. Angaben zum Ziel:

AKKU-1-L

Nummer des ersten zu beschreibenden Datenwortes im aufgeschlagenen Datenbaustein:

Die zulässigen Werte orientieren sich an der Länge des Datenbausteins im Speicher. Es können dabei Nummern > 255 auftreten.

Ergebnis

Wird der Sonderfunktions-OB 190/192 korrekt bearbeitet, so wird das VKE gelöscht (VKE = 0). Die AKKUs bleiben unverändert.

Im Fehlerfall wird das VKE gesetzt (VKE = 1), die AKKUs bleiben unverändert.

Fehlerfälle

- Kein DB- oder DX-Datenbaustein aufgeschlagen,
- falscher Merkerbereich (Letztes Merkerbyte < Erstes Merkerbyte),
- Datenwort-Nummer nicht vorhanden,
- Länge des DB- oder DX-Datenbaustein nicht ausreichend.

OB 191/193: Datenblöcke in Merkerbereich übertragen

Anwendung

Mit Hilfe der Organisationsbaustein OB 191 und OB 193 können sie Daten aus einem Datenbaustein in den Merkerbereich übertragen. So können beispielsweise die zuvor in einem Datenbaustein "geretteten" Merkerbytes wieder in den Merkerbereich zurückgeschrieben werden.

OB 191/193 unterscheiden sich von den Organisationsbausteinen OB 190/192 nur dadurch, dass Quelle und Ziel vertauscht sind:

OB 190/192:	Merkerbereich	→ Datenbaustein
OB 191/193	Merkerbereich ←	Datenbaustein

Hinweis!

Vor dem Aufruf von OB191/193 muss ein ausreichend langer Datenbaustein (DB/DX) aufgeschlagen werden!

Die OB 191/193 übertragen aus dem Datenbaustein nur in den M-Merkerbereich, jedoch nicht in den S-Merkerbereich.

Funktion

Nach Aufruf des OB191/193 werden aus dem aufgeschlagenen Datenbaustein Datenwörter ab der angegebenen Datenwortadresse gelesen und in den Merkerbereich übertragen.

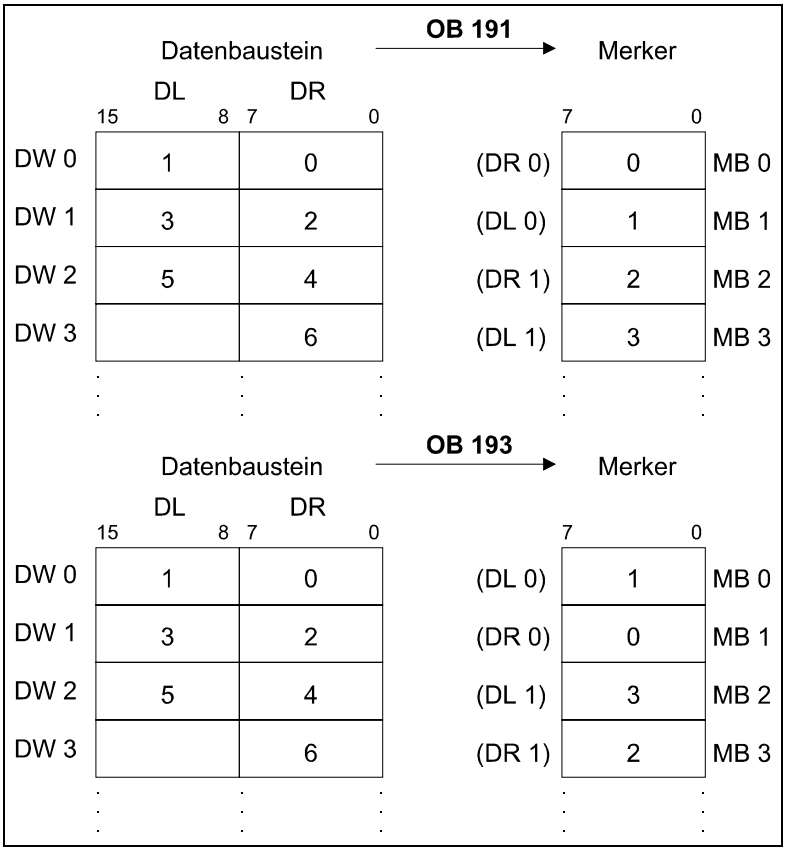
OB 191 und OB 193 sind identisch mit Ausnahme der Art und Weise, in der sie die Daten übertragen:

OB 191 überträgt die Merker **byteweise**.

OB 193 überträgt die Merker **wortweise**.

Die folgende Abbildung soll diesen Unterschied verdeutlichen:

Byteweises und wortweises Übertragen



Parameter

1. Angaben zur Datenquelle:

AKKU-2-L

Nummer des ersten zu übertragendes Datenwortes im aufgeschlagenen Datenbaustein.

2. Angaben zum Ziel:

AKKU-1-LH

Erstes zu beschreibendes Merkerbyte,

zulässige Werte: 0 bis 255.

AKKU-1-LL

Letztes zu beschreibendes Merkerbyte,

zulässige Werte: 0 bis 255 (**Letztes Merkerbyte ≥ Erstes Merkerbyte**).

Ergebnis

Wird der Sonderfunktions-OB 191/193 korrekt bearbeitet, so wird das VKE gelöscht (VKE=0). Die AKKUs bleiben unverändert.

Im Fehlerfall wird das VKE gesetzt (VKE = 1), die AKKUs bleiben unverändert.

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- kein DB- oder DX-Datenbaustein aufgeschlagen,
- falscher Merkerbereich (Letztes Merkerbyte < Erstes Merkerbyte),
- Datenwort-Nummer nicht vorhanden,
- Länge des DB- oder DX-Datenbaustein nicht ausreichend.

Beispiele

Beispiel 1:

Vor Aufruf des Programmbausteins PB 12 sind alle Merker (MB 0 bis MB 255) in den Datenbaustein DX 37 ab Adresse 100 zu retten und anschließend wieder zurückzuschreiben.

```
Retten:
          :AX     DX37      Datenbaustein aufschlagen
          :L      KY0,255   Merkerbereich MB 0 bis MB 255
          :L      KB100     Nummer des 1. Datenwortes im Ziel
          :SPA    OB190     Merker retten

Bausteinwechsel:
          :SPA    PB12

Zurückschreiben:
          :                 (Datenbaustein ist bereits offen)
          :L      KB100     Nummer des 1. Datenwortes in der
                            Quelle
          :L      KY0,255   Merkerbereich MB 0 bis MB 255
          :SPA    OB191     Merkerbereich zurückschreiben
```

Beispiel 2:

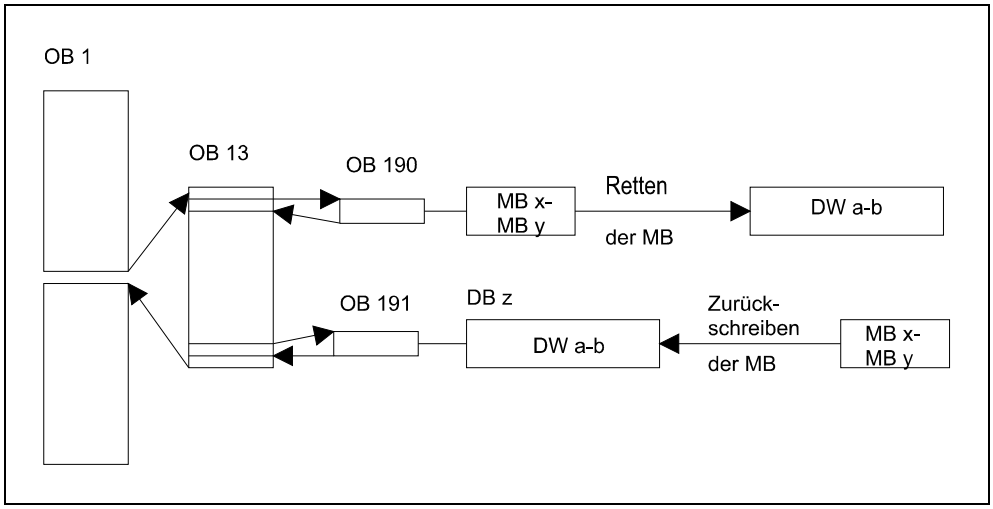
Merker, die vom zyklischen Anwenderprogramm benutzt werden, können nicht zusätzlich durch ein zeit- oder alarmgesteuertes Anwenderprogramm benutzt werden. Jeder Programmbearbeitungsebene muss ein bestimmter Teil des Merkerbereiches zugeordnet sein.

z.B.	zyklisches Anwenderprogramm	MB 0 ..... MB 99
	zeitgesteuertes Anwenderprogramm	MB100.....MB199
	alarmgesteuertes Anwenderprogramm	MB200.....MB255

Falls jedoch das zyklische Anwenderprogramm bereits alle 256 Merkerbytes benutzt und beispielsweise das zeitgesteuerte Anwender-programm ebenfalls alle 256 Merkerbytes benötigt, müssen die Merker beim Wechsel der Bearbeitungsebene ausgetauscht und zwischen-gespeichert werden.

Am schnellsten können die Merker mit Hilfe der Sonderfunktionen OB 190 und OB 191 gerettet und geladen werden. Bild KEIN MERKER zeigt, wie ein von OB 1 und OB 13 (100ms Weckalarm) gemeinsamer Merkerbereich MB X bis MB Y in einem Datenbaustein zwischengespeichert werden:

Retten von Merkerbereichen



Programm im OB 13:

```
:A      DB100
:L      KY0,255
:LK     KB0
:SPA    OB190
:L      KB128
:L      KY0,255
:SPA    OB191
:A      DB100
:L      KY0,255
:L      KB128
:SPA    OB190
:L      KB0
:L      KY0,255
:SPA    OB191
:BE
```

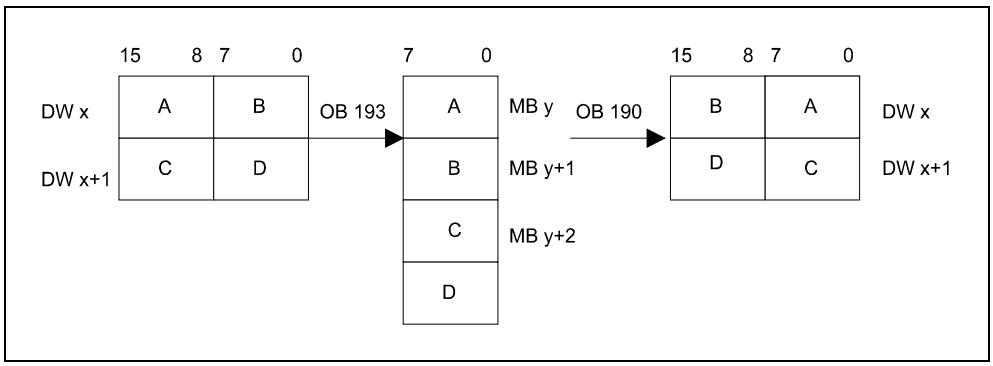
Weitere Anwendungen für die Organisationsbausteine OB 190 bis 193

Bei der CPU 928 werden Operationen zur Einzelbitbearbeitung (U, O, UN, ON, S, R, =), die auf den Merkerbereich zugreifen, wesentlich schneller bearbeitet als vergleichbare Operationen, die auf Datenbausteine zugreifen (vergleichen Sie hierzu z. B. die Operationen 'UM' 'U D' oder 'S M' 'S D').

Aus diesem Grund verbessern Sie die Laufzeit, wenn Sie die Daten in den Merkerbereich kopieren, diese dort bearbeiten und anschließend wieder in den Datenbaustein zurück übertragen.

Ohne großen Aufwand lassen sich im Datenbaustein Hight-Byte und Low-Byte vertauschen, indem mit den entsprechenden OBs Datenwörter in den Merkerbereich und wieder zurück übertragen werden, wie in Bild dargestellt:

Vertauschen von High und Low-Byte



Sie können Datenblöcke innerhalb eines Datenbausteins "verschieben", wenn Sie beim Zurückübertragen aus dem Merkerbereich zwar dieselbe DB-Nummer, jedoch ein anderes Datenwort angeben.

OB 216 bis 218: Kachelzugriffe

Was sind Kacheln?

Um im Adressbereich des AG-Busses eine große Anzahl von Kommunikationsspeichern unterbringen zu können, ist ein Adressbereich mit einer Länge von 1024 Byte (reserviert sind 2048 Byte) 256-mal auf den Speicher abgebildet. Weil diese 256 Abbildungen wie einzelne "Kacheln" neben- oder hintereinander liegen, werden diese Speicherbereiche auch "Kachelspeicher" genannt.

Zur Anwahl der benötigten Kachel dient ein "Kacheladressregister", das auf allen mit Kacheln arbeitenden Baugruppen vorhanden ist und eine feste Adresse auf dem AG-Bus hat. Auf jeder dieser Baugruppen durch den Anwender per DIL-Schalter oder Software die Nummern (Adressen) der Kacheln eingestellt, so dass jede Kachel nur einmal im AG vorhanden ist.

Die CPU gibt vor jedem Lesen oder Schreiben einer Kachel durch Schreiben in das Kachelregister die Kachelnummer an. Alle nach diesem Verfahren arbeitenden Baugruppen auf dem AG-Bus empfangen gleichzeitig ("broadcast") diese Nummer und legen sie in ihrem Speicher ab. Nur die so adressierte Kachel ist nun über den AG-Bus schreib- und lesbar, alle übrigen Kacheln sind gesperrt.

Wie können Sie auf Kacheln zugreifen?

Die Organisationsbausteine OB 216 bis OB 218 sowie einige SPS-Operationen ermöglichen den Zugriff auf sogenannte "Kacheln".

Die Organisationsbausteine enthalten folgende Funktionen:

OB 216

Schreiben eines Bytes/Wortes/Doppelwortes auf eine Kachel,

OB 217

Lesen eines Bytes/Wortes/Doppelwortes von einer Kachel,

OB 218

Belegen einer Kachel durch die CPU.

Diese Funktionen dienen Testzwecken und ermöglichen die Programmierung von Hantierungsbausteinen (FB 120 bis 127) oder ähnlichen Funktionen.

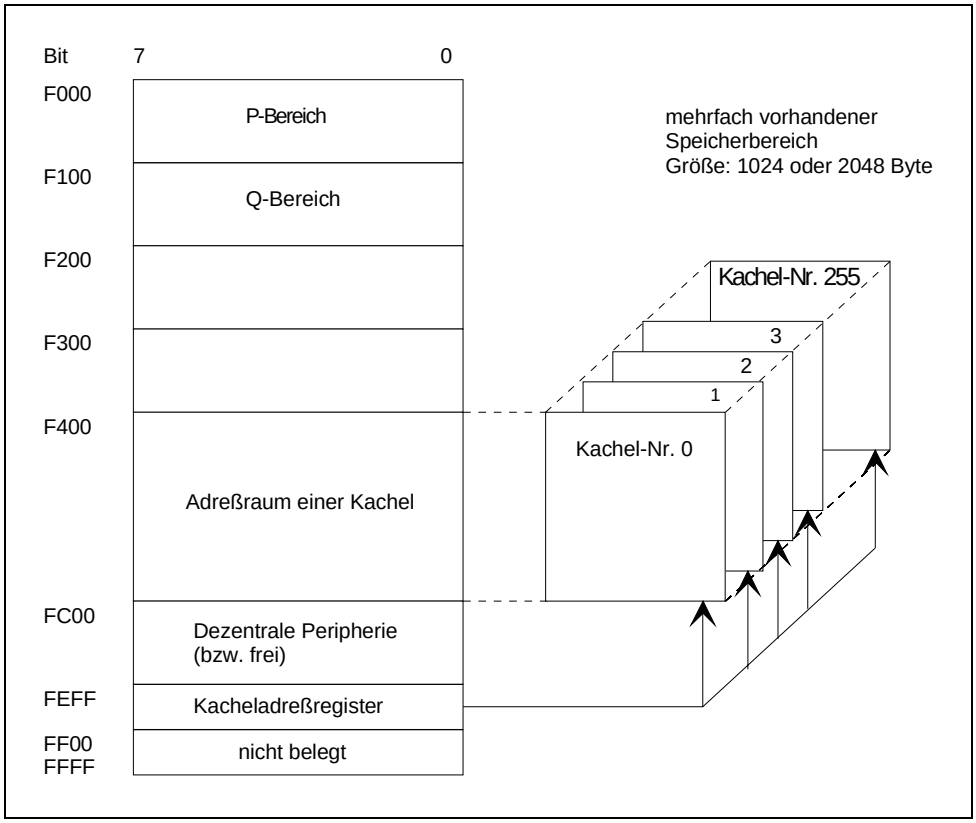
Hinweis!

In der Regel können Sie alle Funktionen mit Hilfe der Standard-Funktionsbausteine "Hantierungsbausteine" (FB 120 bis FB127) ausführen, mit denen alle Kachelzugriffe "automatisch" abgewickelt werden.

Kachelzugriffe sollten Sie möglichst nur durch Aufrufe der OB 216 bis OB 218 programmieren. Die dafür auch zur Verfügung stehenden SPS-Operationen sollten Sie dagegen nur verwenden, wenn Sie über sehr gute Systemkenntnisse verfügen.

Adressbereiche für Peripherie auf dem AG-Bus

Kachelgröße	Belegter Adressraum
1024 Adressen (Byte- oder Wortadressen)	F400h bis F7FFh
2048 Adressen (Byte- oder Wortadressen)	F400h bis FBFFh



Welche der insgesamt 256 Kacheln verwendet werden soll, geben Sie bei der Parametrierung der Sonderfunktions-Organisationsbausteine OB 216, OB 217 und OB 218 an. Die Nummer der "aktuellen" Kachel wird daraufhin automatisch in eine Zelle mit der Adresse 0FEFFh eingetragen

Alle Datenübertragungen beziehen sich dann auf die Kachel, deren Nummer eingetragen wurde.

OB 216: Schreiben auf eine Kachel

Funktion

Der Sonderfunktions-Organisationsbaustein überträgt ein Byte, Wort oder Doppelwort vom AKKU 1 (rechtsbündig) zu einer bestimmten Kachel.

Das Adressieren der und das Übertragen des vollständigen Datums (1,2, oder 4 Bytes) bilden eine untrennbare Programmeinheit, die nicht unterbrochen werden kann.

Parameter

Akkus

AKKU-3-LH

Kennung des zu übertragenden Datums,

zulässige Werte: 0 = Byte
1 = Wort
2 = Doppelwort.

AKKU-3-LL

aktuelle Kachel-Nummer,

zulässige Werte: 0 bis 255.

AKKU-2-L

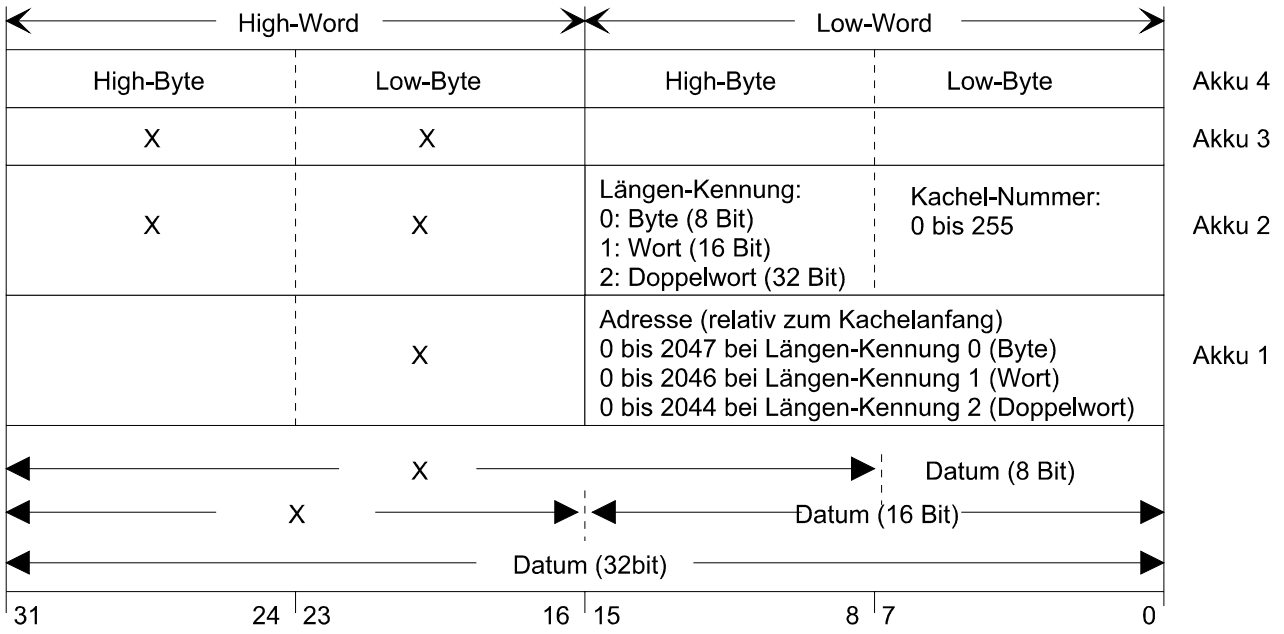
Zieladresse auf der Kachel,

zulässige Werte: 0 bis 2047.

AKKU 1

Datum, das geschrieben werden soll (Byte, Wort oder Doppelwort: rechtsbündig).

Akku-Belegung **vor** dem Schreiben (vor Aufruf des OB 216):



Ergebnis

- Das Schreiben auf die Kachel verläuft erfolgreich:
  AKKU 1 und AKKU 3: werden nicht verändert,
  AKKU-2-L: enthält einen um 1, 2, oder 4 erhöhten Wert (je nach Länge des übertragenen Datums),
  VKE = 1,
  restliche Bit und Wortanzeigen werden gelöscht.

- Das Schreiben auf die Kachel ist nicht möglich:
  alle Akkus: werden nicht verändert
  VKE = 0,
  restliche Bit und Wortanzeigen: werden gelöscht.

Fehlerfälle

Es werden folgend Fehlerfälle unterschieden:

- falsche Längenkennung in AKKU-3-LH,
- Zieladresse auf der Kachel falsch oder nicht vorhanden,
- angegebene Kachel-Nummer nicht vorhanden.

OB 217: Lesen aus einer Kachel

Funktion

Der Sonderfunktions-Organisationsbaustein überträgt ein Byte, Wort oder Doppelwort von einer bestimmten Kachel zu AKKU 1 (rechtsbündig).

Das Adressieren der Kachel und das Übertragen des vollständigen Datums (1, 2 oder 4 Bytes) bilden eine untrennbare Programmeinheit, die nicht unterbrochen werden kann.

Parameter

Akkus

AKKU-3-LH

Kennung des zu übertragenden Datums,

zulässige Werte: 0 = Byte
1 = Wort
2 = Doppelwort

AKKU-3-LL

aktuelle Kachel-Nummer,

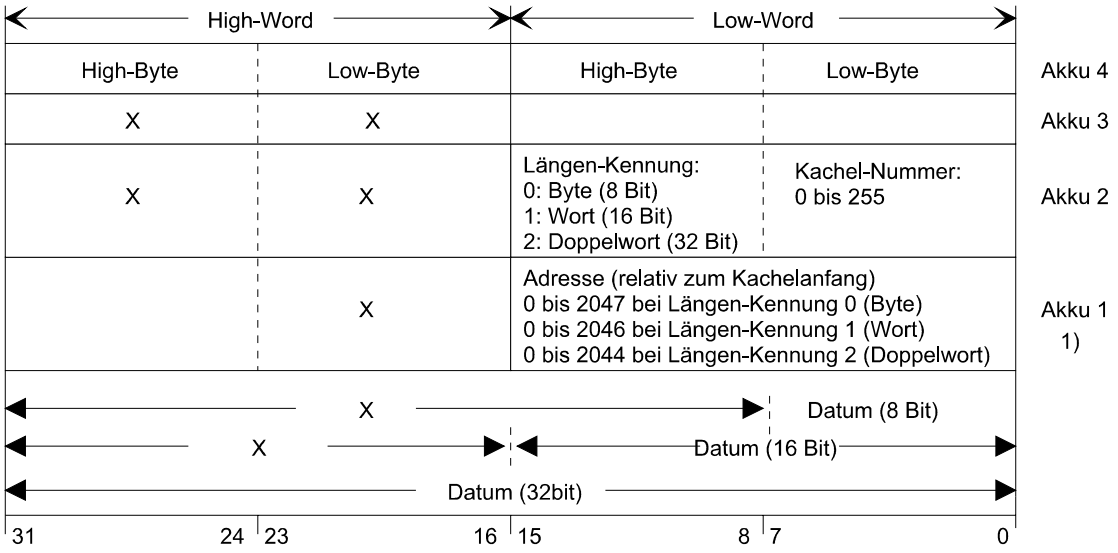
zulässige Werte: 0 bis 255

AKKU-2-L

Quelladresse auf der Kachel,

zulässige Werte: 0 bis 2047

Akku-Belegung **vor** dem Lesen (vor Aufruf des OB 217):



1) Datenablage nach Aufruf des OB 217

Ergebnis

- Das Lesen der Kachel verläuft erfolgreich:

AKKU 1:	enthält (rechtsbündig) den gelesenen Wert (der mögliche Rest der 32 Bits wird gelöscht),
AKKU 3:	wird nicht verändert,
AKKU-2-L:	enthält einen um 1, 2, oder 4 erhöhten Wert (je nach Länge des übertragenen Datums),
VKE:	= 1,
restliche Bit und Wortanzeigen:	werden gelöscht.

- Das Lesen von der Kachel ist nicht möglich:

alle Akkus:	werden nicht verändert,
VKE:	= 0,
restliche Bit und Wortanzeigen:	werden gelöscht

Fehlerfälle

Es werden folgende Fehlerfälle unterschieden:

- falsche Längenkennung im AKKU-3-LH,
- Quelladresse auf der Kachel falsch oder nicht vorhanden,
- angegebene Kachel-Nummer nicht vorhanden.

OB 218: Belegung einer Kachel

Funktion

Der Sonderfunktions-Organisationsbaustein überträgt die Nummer der CPU zu einer bestimmten Kachel, falls der Inhalt der adressierten Zelle auf dieser Kachel gleich Null ist. Solange nun die CPU-Nr. in der Zelle eingetragen bleibt, ist diese Kachel für diese CPU reserviert und kann von anderen CPUs nicht belegt werden.

Der Organisationsbaustein OB 218 dient der Synchronisation des Datentransfers und ist besonders wichtig, wenn größere, zusammengehörige Datenblöcke geschlossen gesendet bzw. übertragen werden sollen.

Das Adressieren der Kachel, das Lesen und das eventuelle Schreiben der Steckplatzkennung bilden eine Programmeinheit, die nicht unterbrochen werden kann.

Parameter

Akkus

AKKU-2-LL

Nummer der zu belegenden Kachel,

zulässige Werte: 0 bis 255.

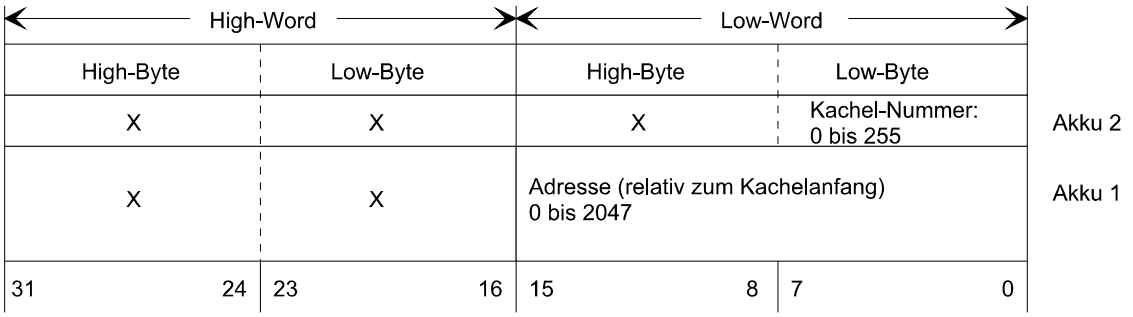
AKKU-1-L

Zieladresse auf der Kachel,

zulässige Werte: 0 bis 2047.

(Die Inhalte von AKKU 3 und 4 sind irrelevant).

Akku-Belegung vor Aufruf des OB 218:



Ergebnis

- Das Belegen der Kachel verläuft erfolgreich:
  alle Akkus: werden nicht verändert,
  VKE: = 1,
  restliche Bit und
  Wortanzeigen werden gelöscht.
- Das Belegen der Kachel ist nicht möglich:
  alle Akkus: werden nicht verändert,
  VKE: = 0,
  restlich Bit
  und Wortanzeigen werden gelöscht.

Fehlerfälle

Folgender Fehlerfall kann auftreten:

- angegebene Kachel-Nummer nicht vorhanden.

OB 220: Vorzeichenerweiterung

Anwendung

Eine Vorzeichenerweiterung ist notwendig, um eine negative 16-bit-Festpunktzahl vor einer Festpunkt-Gleitpunkt-Wandlung (32-bit, Operation FDG) zu einer 32-bit-Festpunktzahl zu erweitern.

Funktion

Diese Sonderfunktion erweitert das Vorzeichen einer 16-bit-Festpunktzahl im AKKU-1-L auf das höherwertige Wort (AKKU-1-H):

- Wenn das Bit 2^{15} =0 (positive Zahl) ist, wird das höherwertige Wort mit KH = 0000 geladen,
- Wenn das Bit 2^{15} =1 (negative Zahl ist, wird das höherwertige Wort mit KH = FFFF geladen.

Parameter

AKKU-1-L

16-bit-Festpunktzahl.

Ergebnis

AKKU-1-H wird entsprechend dem Vorzeichen der Festpunktzahl in AKKU-1-L geladen (s.o.).

Fehlerfälle

keine

OB 221: Zyklusüberwachung einstellen

Funktion

Durch Aufruf dieser Sonderfunktion können Sie die Zyklusüberwachungszeit ändern und damit die maximal zulässige Zykluszeit neu festlegen. Standardmäßig ist die Zyklusüberwachungszeit auf 150ms eingestellt.

Mit diesem Aufruf wird gleichzeitig der Timer für die Überwachung neu gestartet. Die maximal zulässige Zykluszeit für den Zyklus, in dem der OB 221 aufgerufen wird, verlängert sich um den neu eingestellten Wert, vom Zeitpunkt des Sonderfunktionsaufrufes an gerechnet. Die Zyklusüberwachungszeit aller folgender Zyklen entspricht dem neu eingestellten Wert (=dem Zeitwert, den Sie im AKKU 1 übergeben).

Parameter

AKKU 1

AKKU-1-L

Neue Zykluszeit (in Millisekunden)

zulässige Werte:	1ms bis 13 000ms
	positive Festpunktzahl (KF)

AKKU-1-H

AKKU-1-H muss den Wert '0' haben

Ergebnis

Nach fehlerfreier Bearbeitung des OB 221 ist die neue Zyklusüberwachungszeit eingestellt.

Fehlerfall

Die angegebene Zyklusüberwachungszeit liegt nicht im Bereich 1 ms bis 13000 ms.

Die Funktion wird nicht ausgeführt. Das Systemprogramm erkennt einen Laufzeitfehler und ruft den OB 31 auf. Die weitere Fehlerreaktion hängt von der Programmierung des OB 31 ab. Ist der OB 31 nicht geladen, geht die CPU in den Stoppzustand.

In beiden Fällen wird in AKKU-1-L die Fehlerkennung 1A3Ah hinterlegt.

OB 222: Zyklusüberwachungszeit neu starten

Funktion

Die Sonderfunktion OB 222 bewirkt ein Nachtriggern der Zyklusüberwachungszeit, d.h. der Timer für die Überwachung wird neu gestartet. Durch Aufruf dieser Sonderfunktion wird die maximal zulässige Zykluszeit für den aktuellen Zyklus um den eingestellten Wert vom Zeitpunkt des Aufrufs an verlängert.

Parameter und Fehlerfälle

keine

OB 228: Statusinformation einer Programmebene lesen

Funktion

Beim Auftreten von bestimmten Ereignissen ruft das Systemprogramm die dazugehörige Programmbearbeitungsebene auf. Die Programmbearbeitungsebene ist damit "aktiviert".

Mit Hilfe des Organisationsbausteins OB 228 können Sie feststellen, ob zu einem Zeitpunkt eine bestimmte Programmbearbeitungsebene aktiviert ist oder nicht. Im AKKU 1 übergeben Sie die Nummer derjenigen Programmbearbeitungsebene, deren Status abgefragt werden soll. (Die Nummern entsprechen den im USTACK unter EBENE eingetragenen Nummern.)

Bei Aufruf des Bausteins hinterlegt dieser die Statusinformation der angegebenen Programmebene im AKKU-1-L. Durch Auswertung diese Information können Sie Ihre Programmbearbeitung abhängig machen vom Status einer anderen Programmbearbeitungsebene.

Parameter

AKKU-1-L

Nummer der Programmbearbeitungsebene (siehe USTACK, EBENE),

zulässige Werte (hexadezimal): siehe nachfolgende Tabelle

Ebenen-Nr. in AKKU-1-L	Ebenen-Name	Ebenen-Nr. in AKKU-1-L	Ebenen-Name
02	NEUSTART	26	nicht belegt
04	ZYKLUS	28	nicht belegt
06	WECKALARM 5s	2A	nicht belegt
08	WECKALARM 2s	2C	Abbruch
0A	WECKALARM 1s	2E	Schnittstellenfehler
0C	WECKALARM 500ms	30	Weckfehler
0E	WECKALARM 200ms	32	Reglerfehler
10	WECKALARM 100ms	34	Zyklusfehler
12	WECKALARM 50ms	36	nicht belegt
14	WECKALARM 20ms	38	Befehlscodefehler
16	WECKALARM 10ms	3A	Laufzeitfehler
18	ZEITAUFTRAG	3C	Adressierfehler
1A	nicht belegt	3E	Quittungsverzug
1C	REGLERALARM	40	nicht belegt
1E	nicht belegt	42	nicht belegt
20	VERZÖGERUNGS-ALARM	44	MANUELLER WIEDERANLAUF
22	nicht belegt	46	AUTOMATISCHER WIEDERANLAUF
24	PROZESSALARM		

Ergebnis

AKKU-1-L:

enthält die Statusinformation:

- gleich 0: Programmbearbeitungsebene ist nicht aufgerufen,
- ungleich 0: Programmbearbeitungsebene ist aktiviert.

AKKU-2-L:

- enthält den vorherigen Inhalt des AKKU-1-L;
- der vorherige Inhalt von AKKU-2-L geht verloren.

Fehlerfälle

keine

Beispiel

Ein Quittungsverzug soll im NEUSTART ignoriert werden, nicht jedoch in den übrigen Programmbearbeitungsebenen.

Sie rufen zu Beginn der Fehlerbearbeitung im OB 23 (QVZ) den OB 228 auf, um festzustellen, ob beim Auftreten des QVZ die Programmbearbeitungsebene NEUSTART (Nummer 02) aktiviert ist oder nicht. Die weitere Fehlerbehandlung machen Sie abhängig von der Statusinformation, die Sie erhalten:

`AKKU-1-L = 0: NEUSTART passiv`	`QVZ ist nicht im NEUSTART, sondern in einer anderen Programmbearbeitungsebene aufgetreten. Das Fehlerprogramm muss bearbeitet werden.`
`AKKU-1-L ≠ 0:NEUSTART aktiviert`	`QVZ ist im NEUSTART aufgetreten, daher darf es ignoriert werden.`

Der OB 228 ermöglicht Ihnen somit u. a. eine differenzierte Fehlerbehandlung.

OB 230 bis 237: Funktionen für Standard-FBs

Einleitung

Die Sonderfunktions-Organisationsbausteine OB 230 bis OB 237 sind für Hantierungsfunktionen reserviert und können nur innerhalb der Standard-Funktionsbausteine FB 120 bis FB 127 aufgerufen werden.

Hantierungsbausteine

Diese Standard-Funktionsbausteine - die sog. "Hantierungsbausteine" - steuern den Datenverkehr über den Kachelbereich: Sie werden eingesetzt, wenn Daten oder Parameter sowie Steuerungsinformationen von den Kommunikationsprozessoren übergeben werden sollen.

Zuordnungshilfe

Der nachfolgenden Tabelle können Sie entnehmen, von welchen Hantierungsbausteinen die Sonderfunktions-Organisationsbausteine
OB 230 bis OB 237 aufgerufen werden.

Standard-Funktionsbaustein	Sonderfunktion-Organisations-baustein	Hantierungsbaustein
FB 120	SF-OB-230	SEND
FB 121	SF-OB-231	RECEIVE
FB 122	SF-OB-232	FETCH
FB 123	SF-OB-233	CONTROL
FB 124	SF-OB-234	RESET
FB 125	SF-OB-235	SYNCHRON
FB 126	SF-OB-236	SEND ALL
FB 127	SF-OB-237	RECEIVE ALL

Anwendung der Hantierungsbausteine

Die Betriebsanleitung, der Hantierungsbausteine, die als Software-Produkt auf Diskette mitgeliefert werden, finden Sie im nächsten Kapitel.

OB 250/251: Regelung /PID-Algorithmus

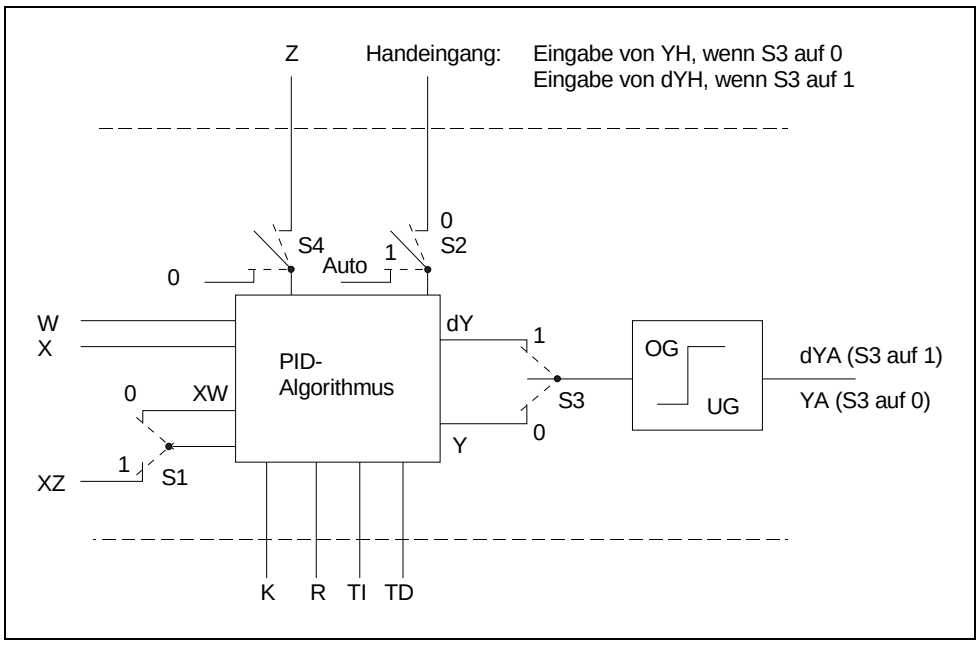
Einleitung

In der CPU 928 können Sie einen oder mehrere PID-Regler aufrufen. Jeder Regler muss im Anlauf-OB initialisiert werden. Zur Übergabe von Parametern wird ein Datenbaustein verwendet.

Der eigentliche Regel-Algorithmus ist im Systemprogramm integriert und vom Anwender lediglich als Organisationsbaustein aufrufbar. Als Datenschnittstelle zwischen Regel-Algorithmus und dem Anwenderprogramm dient ein Datenbaustein.

Funktionsbeschreibung des PID-Reglers

Blockschaltbild des PID-Reglers



Index k

k-te Abtastung

Schalter	Stellung	Wirkung
S1 STEU-BIT1	0	Dem Differenzierer wird die Regeldifferenz XW_k zugeführt.
	1	Dem Differenzierer kann über XZ ein anderes Signal zugeführt werden.
S2 STEU-BIT0	0 1	Handbetrieb Automatik
S3 STEU-BIT3	0 1	Stellungs-Algorithmus Geschwindigkeits-Algorithmus
S4 STEU-BIT5	0 1	Mit Störungsaufschaltung Ohne Störungsaufschaltung

Steuerwort STEU

Durch entsprechendes Setzen der Steuerbits können Sie die dem Blockschaltbild entsprechende Regler-Funktion beeinflussen.

Der kontinuierliche Regler ist für schnelle Regelstrecken, wie sie z. B. in der Verfahrenstechnik als Druck-, Temperatur- oder Durchflussregelungen auftreten, ausgelegt.

PID-Algorithmus

Dem Regler selbst liegt ein PID-Algorithmus zugrunde. Sein Ausgangssignal kann wahlweise als Stellgröße (Stellungs-Algorithmus) oder als Stellgrößenänderung (Geschwindigkeits-Algorithmus) ausgegeben werden.

Die einzelnen P- ,I- und D-Anteile sind über ihre jeweiligen Parameter R, TI und TD abschaltbar, indem die betreffenden Zellen mit Null vorbesetzt werden. Damit können alle gewünschten Reglerstrukturen, z. B. PI-, PID- oder PD-Regler, leicht realisiert werden.

Differenzierer

Dem Differenzierer kann wahlweise die Regeldifferenz XW oder eine beliebige Störgröße (über den XZ-Eingang) oder der invertierte Istwert -x zugeführt werden.

Störkompensation

Für den Fall, dass zur Kompensation eines Störgrößeneinflusses eine Vorsteuerung des Stellgliedes ohne Zeitverhalten erwünscht ist, kann dem Regel-Algorithmus eine im Prozess messtechnisch erfassbare Störgröße Z aufgeschaltet werden. Im Handbetrieb wird an dieser Stelle die vorgewählte Stellgröße YN übernommen.

Invertierter Regelsinn

Wenn ein invertierter Reglersinn gefordert wird, ist ein negativer K-Wert vorzugeben.

Begrenzung der Stellinformation

Wenn die Stellinformation (dY oder Y) an einer Begrenzung liegt, wird der I-Anteil automatisch abgeschaltet, um eine Verschlechterung des Reglerverhaltens zu vermeiden.

Das Reglerprogramm kann durch Vorgabe von Festwerten oder über adaptive (dynamische) Vorgabe von Parametern (K, R, TI, TD) versorgt werden. Die Eingabe erfolgt über die den einzelnen Parametern zugeordneten Speicherzellen.

PID-Algorithmus

Einleitung

Dem PID-Regler liegt ein Geschwindigkeits-Algorithmus zugrunde, nach dem zu einem bestimmten Zeitpunkt t = k TA das jeweilige Stellinkrement dY_k nach folgender Formel berechnet wird:

$$dY_k = K[(XW_k - XW_{k-1})R + \frac{TA}{2TN}(XW_k + XW_{k-1}) + \frac{1}{2}\left\{\frac{TV}{TA}(XU_k - 2XU_{k-1} + XU_{k-2}) + dD_{k-1}\right\}]$$

$$= K(dPW_kR + dI_k + dD_k)$$

P-Anteil I-Anteil D-Anteil

dXX_k:Änderung der Größe XXX zum Zeitpunkt t.

U kann W oder Z sein, je nachdem, ob die Differenzierer XW oder XZ zugeführt wird. Entsprechend gilt:

Bei XW_k-Zuführung	Bei XZ-Zuführung
$PW_k = W_k - X_k$	
$PW_k = XW_k - XW_{k-1}$	$PZ_k = XZ_k - XZ_{k-1}$
$QW_k = PW_k - PW_{k-1}$	$QZ_k = PZ_k - PZ_{k-1}$
$QW_k = XW_k - 2XW_{k-1} + XW_{k-2}$	$QZ_k = XZ_k - 2XZ_{k-1} + XZ_{k-2}$

$dPW_k = (XW_k - XW_{k-1})R$

$dIk_k = TI \cdot XW_k \qquad TI = \frac{TA}{TN}$

$dD_k = \frac{1}{2}(TD \cdot QU_k + dQ_{k-1}) \quad TD = \frac{TV}{TA}$

Wenn als Reglerausgang zum Zeitpunkt t_k die Stellgröße Y_k gewünscht wird, wird sie nach folgender Formel gebildet:

$$Y_k = \sum_{m=0}^{m=k} dY_m$$

Bei den meisten Reglerentwurfsverfahren geht man davon aus, dass R =1 ist, wenn ein P-Verhalten erwünscht ist.

Mit der Größe R kann der Proportionalanteil des PID-Reglers eingestellt werden.

Datenbausteine für den PID-Regler

Die reglerspezifischen Daten werden mit Hilfe eines Übergabe-Datenbausteins eingegeben (Initialisierung und Bearbeitung des PID-Reglers siehe entsprechende Abschnitte diese Kapitels).

Diese Daten müssen Sie im Übergabe-Datenbaustein x vorgeben:

K, R, TI, TD, W, STEU, YH, BGOG, BGUG.

Der Übergabe-Datenbaustein muss aus 49 Datenwörtern mit den Nummern 0 bis 48 bestehen.

Die Datenbelegung dieser Datenwörter wird in der nachfolgenden Tabelle erläutert.

Aufbau des Übergabe-Datenbausteins

Adr. im DB	Name	E/A 1)	Zahlen-format 2)	PG-Format 3)	Bemerkung
DW0	-	-	-	-	Reserve
DD1	K	E	GP	KG	Proportionalbeiwert K>0: Pos.Regelsinn, d.h. gleichsinnige Änderung von Sollwert und Stellgr. K<0: Neg. Regelsinn, Gleitpunktzahl
DD3	R	E	GP	KG	R-Parameter=1 bei Reglern mit P-Anteil
DD5	TI	E	GP	KG	TI=TA/TN
DD7	TD	E	GP	KG	TD=TV/TA
DD9	W_k	E	GP	KG	Sollwert-Eingabe hier,wenn STEU-Bit 6=1,ansonsten in Wort Nr. 19 (-1≤YWk <1)
DW11	STEU	E	BM	KM	Steuerwort
DD12	YH_k	E	GP	KG	Handwert-Eingabe hier, wenn STEU Bit 6=1, ansonsten in Wort Nr. 18 (-1≤Yhk<1). Bei Geschw.-Algorithmus sind hier Stellwert-Inkremente anzugeben
DD14	BGOG	E	GP	KG	Oberer Begrenzungsw. 4) -1≤BGOG≤1 ($YA_{k\,max}$), BGUG<BGOG !!!
DD16	BGUG	E	GP	KG	Unterer Begrenzungsw. 4) -1≤BGUG≤1 $YA_{k\,max}$)
DW18	Yh_k	E	LP	KF	Handwert-Eingabe hier, wenn STEU-Bit 6=0 (-1≤YH<1) Bei Geschw. Algorithmus sind hier Stellwert-Inkremente anzugeben.

Fortsetzung ...

Adr. im DB	Name	E/A [1]	Zahlen-format [2]	PG-Format [3]	Bemerkung
DW19	W_k	E	LP	KF	Sollwert-Eingabe hier, wenn STEU-Bit 6=0 (-1≤Wk<1)
DW20	MERK	-	BM	KF	Bit 0 = 1: pos. Begrenzung überschritten Bit 1 = 1: neg. Begrenzung unterschritten
DW21	X_k	E	LP	KF	Istwert-Eingabe für STEU-Bit 7=0 (-1≤X_k<1)
DD22	X_k	E	GP	KG	Istwert-Eingabe für STEU-Bit 7=1 (-1≤X_k<1)
DW24	Z_k	E	LP	KF	Störgröße (-1≤Z_k<1) für STEU-Bit 7=0
DD25	Z_k	E	GP	KG	Störgrößeneingabe wenn STEU-Bit 7=1 (-1≤Z_k<1)
DD27	Z_{k-1}	-	GP	KG	Vergangenheitswert der Störgröße
DW29	XZ_k	E	LP	KF	Über den Eingang XZ dem Differenzierer zugef. Größe (-1≤XZ_k<1) Eingabe hier, wenn STEU-Bit 7=0
DD30	XZ_k	E	GP	KG	XZ-Eingabe hier, wenn STEU-Bit 7=1 (-1≤XZ_k<1)
DD32	XZ_{k-1}	-	GP	KG	Vergangenheitswert von XZ_k
DD34	PZ_{k-1}	-	GP	KG	XZ_{k-1} - XZ_{k-2}
DD36	dD_{k-1}	-	GP	KG	Differentialanteil
DD38	XW_{k-1}	-	GP	KG	Vergangenheitwert der Regeldifferenz
DD40	PW_{k-1}	-	GP	KG	XW_{k-1}-XW_{k-2}
DW42	-	-	GP	KG	Reserve
DD44	Y_{k-1}	-	GP	KG	Vergangenheitswert der berechneten Stellgröße Y_{k-1} bzw. dY_{k-1} vor dem Begrenzer
DD46	YA_k	A	GP	KG	Ausgangsgröße
DW48	YA_k	A	LP	KF	Ausgangsgröße BGUG≤YA≤BGOG

[1] E = Eingabe, A = Ausgabe

[2] GP = Gleitpunktzahl, LP = Linkspunktzahl, BM = Bitmuster

[3] Vorgeschlagenes Format (KH, KM ebenfalls zulässig)

[4] Im Linkspunktformat müssen oberer und unterer Begrenzungswert nach folgenden Formeln eingegeben werden:

DD 14 = BGOG: Wert als Gleitpunktzahl = BGOG/32767

DD 16 = BGUG: Wert als Gleitpunktzahl = BGUG/32767

Beispiel für Begrenzungswerte

Begrenzungswerte

Oberer Begrenzungswert = 0,1

Unterer Begrenzungswert = -0,1

Einträge in den DB:

DD 14: +1000 000 +00

DD 16 : -1000 000 +00

Ausgangsgröße wird begrenzt:

DW 48: ±3276

DW 46: ±0,1

Bemerkung

Für die Begrenzungswerte außerhalb 1 wird die Ausgangsgröße im GP-Format begrenzt (DD 46).

Belegung des Steuerwortes STEU (Datenwort DW 11 im Übergabe-DB)

DW 11 Bit-Nr.	Name	Bedeutung
11.0	AUTO	= 1: Automatikbetrieb = 0: Handbetrieb
11.1	XZ_EIN	= 1: Dem Differenzierer wird über den XZ-Eingang eine andere Größe zugeführt, die nicht XW_k sein darf. = 0: Dem Differenzierer wird XW_k zugeführt. Der XZ-Eingang bleibt unberücksichtigt.
11.2	REG_AUS	= 1: Beim Aufruf des Reglers (OB 251) werden mit Ausnahme von K, R, TI, TD, BGOG, BGUG, STEU,Yh_k,W_k, Z_k und Z_{k-1} alle anderen Größen (DW20 bis DW48) im Regler-DB einmal gelöscht. Der Regler ist ausgeschaltet. Der Vergangenheitswert der Störgr. wird aktualisiert. = 0: Regeln
11.3	GESCHW	=1: Geschwindigkeits-Algorithmus =0: Stellungs-Algorithmus
11.4	HAND-ART	=1: Bei GESCHW =0 wird die zuletzt ausgegebene Stellgröße beibehalten. Bei GESCHW =1 wird das Stellinkrement dY_k =0 gesetzt. =0: Bei GESCHW = 0 wird nach dem Umschalten auf Handbetrieb der ausgegebene Stellwert YA in 4 Abtastschritten exponentiell auf den eingestellten Handwert geführt. Danach werden weitere Handwerte sofort am Reglerausgang übernommen. Bei GESCHW = 1 werden die Handwerte sofort auf den Relerausgang durchgeschaltet. Im Handbetrieb sind die Begrenzungen wirksam. Im Handbetrieb werden folgende Größen aktualisiert: X_k, XW_{k-1},und PW_{k-1} X_k,XZ_{k-1} und PZ_{k-1}, wenn STEU-Bit 1 = 1 Z_k, Z_{k-1}, wenn STEU-Bit 5 = 0 Die Größe dD_{k-1} wird = 0 gesetzt. Der Algorithmus wird nicht abgearbeitet.
11.5	NO_Z	=1: keine Störgrößenaufschaltung =0: mit Störgrößenaufschaltung
11.6	PGDG	=1: W_k, Yh_k-Eingabe als Gleitpunktzahl =0: Eingabe als Linkspunktzahl
11.7	VAR_GP	=1: Die Variablen X_k, XZ_k und Z_k werden als Gleitpunktzahlen eingegben. =0: Eingabe der Variablen als Linkspunktzahl
11.8	STOS	=1: keine stoßfreie Hand-Automatik-Umschaltung =0: stoßfreie Hand-Automatik-Umschaltung
11.9 bis 1.15		ohne Bedeutung

OB 250: PID-Algorithmus initialisieren

Funktion

Der OB 250 initialisiert den PDI-Algorithmus und wird in den Anlauf-OBs 20, 21, 22 aufgerufen.

Parameter

Die für das Initialisieren erforderlichen Parameter stehen im Übergabe-Datenbaustein (DBx).

Hinweis!

Der Übergabe-Datenbaustein muss vor Aufruf des OB 250 aufgeschlagen werden.

Für jeden Regler muss zur Datenübergabe ein eigener DB x verwendet werden (x≤254). Das Systemprogramm erzeugt daraus automatisch einen weiteren DBx+1 im Datenbaustein-RAM, den der Regler im zyklischen Betrieb als Datenfeld verwendet; die entsprechenden DB-Nummern müssen also noch frei sein. Die Datenbausteine DBx+1 sind die jeweilige Datenschnittstelle zwischen den Reglern und dem Anwender bzw. der Peripherie.

Fehlerfälle

Der OB 250 verwendet intern den OB 254 bzw. OB 255 (Duplizieren von Datenbausteinen).

Im Fehlerfall erkennt die CPU einen Laufzeitfehler und ruft den OB 31 auf. Wenn dieser nicht programmiert ist, geht die CPU in den Stoppzustand. Die im AKKU 1 hinterlegten Fehlerkennungen beziehen sich dann auf den OB 250.

Achtung!

Wenn bei der Initialisierung der DBx+1 nicht freigehalten war, wird dieser ohne Meldung vom Systemprogramm als Reglerdatenfeld verwendet, sofern er die gleiche Länge hat wie ein Regler-DB (48 Datenwörter); dabei werden die Datenwörter 20 bis 48 gelöscht.

Ansonsten geht die CPU in den Stoppzustand.

Statt Datenbausteinen DB können auch erweiterte Datenbausteine DX verwendet werden. Die Initialisierung verläuft dabei analog zu der bei Datenbausteinen DB.

OB 251: PID-Algorithmus bearbeiten

Anwendung

Der OB 251 wird während der zyklischen Programmbearbeitung aufgerufen und bearbeitet den PID-Algorithmus.

Aufruf

Nach Ablauf der Abtastzeit soll der Regler aufgerufen werden. Halten Sie dabei folgende Reihenfolge ein:

1. Datenbaustein DB x+1 aufschlagen
2. Eingangsdaten X_k, XZ_k,Z_k und YH_k oder eine Untermenge davon laden
3. Eingangsdaten formrichtig umwandeln und in den DB x+1 transferieren
4. OB 251 (PID-Regler bearbeiten) aufrufen
5. Ausgangsdatums YA_k aus dem DB x+1 laden
6. Datum umwandeln und zur Prozessperipherie transferieren

Format der Reglereingänge und -ausgänge

Der PID-Regelalgorithmus verwendet intern zur Zahlendarstellung das Gleitpunktformat und kann mit Gleitpunktwerten versorgt werden. Eine Versorgung des PID-Regelalgorithmus im Linkspunktformat ist ebenfalls möglich (siehe dazu Bit 6 und 7 im Steuerwort STEU). In diesem Fall wandelt der Regler bei jedem Aufruf selbständig die Wörter ins Gleitpunktformat um.

Die Anpassung der Wörter von den Eingabe- und Ausgabebaugruppen im Anwender-Programm ist laufzeitgünstiger bei Verwendung des Linkspunktformates (siehe Tabelle am Ende dieses Kapitels).

Eingänge

W, YH, X, Z und XZ können wahlweise als Gleit- oder Linkspunktzahl eingegeben werden. Im Datenübergabebaustein sind für jede Größe jeweils unterschiedliche Speicherplätze vorgesehen.

Eingabe als Linkspunktzahl

Erläuterungen zur Linkspunktzahl: Siehe Tabelle am Ende dieses Kapitels.

Hinweis!

Bei Einhaltung der Eingangsnennbereiche der Analogeingabebaugruppen müssen Sie berücksichtigen, dass das Bitmuster für einen bestimmten Eingangswert anders ist als bei Ausnutzung des vollen Eingangsbereichs. Dies ist insbesondere bei der Sollwerteinstellung sehr wichtig, da es sonst vorkommen kann, dass ein über das PG eingegebener Sollwert nicht erreicht werden kann, obwohl der Istwert weit über dem gewünschten Wert liegt.

Wenn der eingesetzte Analog-Digital-Umsetzer die negativen Zahlen als Betrag als Betrag und Vorzeichen liefert, muss man daraus, bevor sie in den Regler-DB transferiert werden, das Zweierkomplement bilden. Anschließend muss die Binärstelle 15 = 1 gesetzt werden.

Wenn die Zahl -0 als Betrag und Vorzeichen in der Form

1000000000000000

beim verwendeten Analog-Digital-Umsetzer möglich ist, darf davon kein Zweierkomplement gebildet werden, sondern die Zahl muss als +0 in den Regler-DB gelangen:

0000000000000000

Ausgang

Der Reglerausgang YA liegt im DB als Links- und als Gleitpunktzahl vor. Linkspunkteingänge und -ausgänge müssen unter Berücksichtigung der verwendeten Ein- und Ausgabebaugruppen (Analog-Digital-Umsetzer, Digital-Analog-Umsetzer) vor und nach dem Regleraufruf im Anwenderprogramm formatgewandelt werden, bevor sie in den bzw. aus dem Regler-DB transferiert werden.

Allgemeine Hinweise

Verwendung von STOS

Wenn STOS (STEU-Bit 8) auf Null steht, verläuft die Umschaltung von Hand- auf Automatikbetrieb stoßfrei, d. h. eine anstehende, beliebig große Regeldifferenz wird über den I-Anteil ausgeregelt. Wenn jedoch TI = TA/TN = 0 gewählt wird (P- oder PD-Regler), verursacht die Regeldifferenz bei der Umschaltung keine Änderung der Stellgröße.

Dies kann vermieden werden, indem man STOS =1 setzt. Eine Regeldifferenz wird dann bei Hand-Automatik-Umschaltung schnell ausgeregelt, gleichgültig ob TI = 0 ist oder nicht. Der dabei entstehende Stellgrößensprung entspricht der Größe der Regeldifferenz, ist also nicht willkürlich im Sinn einer Störung des Reglerbetriebes.

Anzeige von MERK, Bit 0 und 1

Bit 0 und 1 von MERK können, falls gewünscht, zur Anzeige gebracht werden, um anzuzeigen, dass die Stellgröße (bei Geschwindigkeitsalgorithmus das Stellinkrement) in der oberen oder unteren Begrenzung ist. Da diese Bits vom Algorithmus zur Abschaltung des I-Anteils ausgewertet werden, dürfen sie nicht überschrieben werden.

Achtung!

Regler-Datenbausteine DBx+1 dürfen während des zyklischen Betriebes nicht neu geladen werden.

Kaskadenregelung

Wenn mit zwei oder mehr Reglern eine Kaskadenregelung aufgebaut wird, ist folgendes zu beachten:

- Wenn die Kaskade aufgetrennt wird, müssen entweder alle Regler gleichzeitig in den Handbetrieb gehen, damit kein Regler infolge seines I-Anteils abdriften kann, oder es muss zumindest der Regler des äußeren Kreises im Handbetrieb arbeiten, damit die letzte Stellgröße, die dem Sollwert des inneren Kreises entspricht, beibehalten oder auf einen Sicherheitswert gefahren werden kann.
- Wenn die Kaskade geschlossen werden soll, sollten beide Kreise gleichzeitig oder wenigstens der innere Kreis im Automatikbetrieb arbeiten, damit die Stellgröße des äußeren Kreises als Sollwert übernommen werden kann.

Umschalten auf Handbetrieb

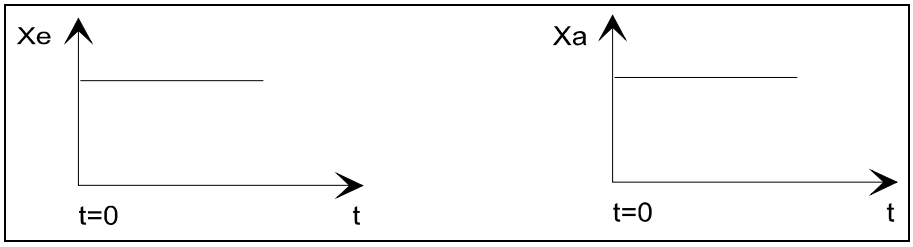
Wenn beim Umschalten auf Handbetrieb die Regelstrecke vom Regler abgetrennt und am Stellglied direkt verstellt wird, ist die so gewonnene Stellgröße über den Handeingang dem Regler zuzuführen. Dies bewirkt dass beim Umschalten von Hand- auf Automatikbetrieb der Reglerausgang mit der im Handbetrieb eingestellten Stellgröße übereinstimmt. Beim Geschwindigkeits-Algorithmus handelt es sich um die Stellgrößenänderung.

Reglerkenngrößen

P-Regler

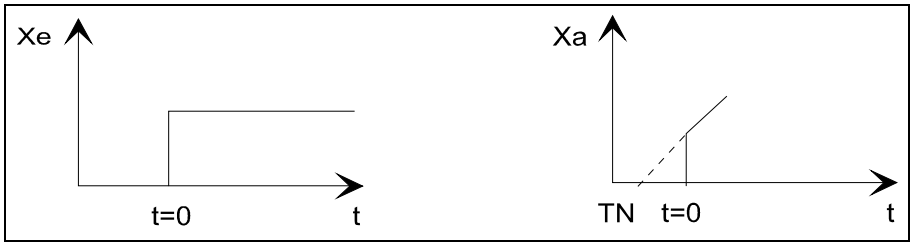
Die Kenngröße für einen P-Regler ist K. Sie ist der Quotient aus Ausgangs- und Eingangsgröße:

K = Xa/Xe.



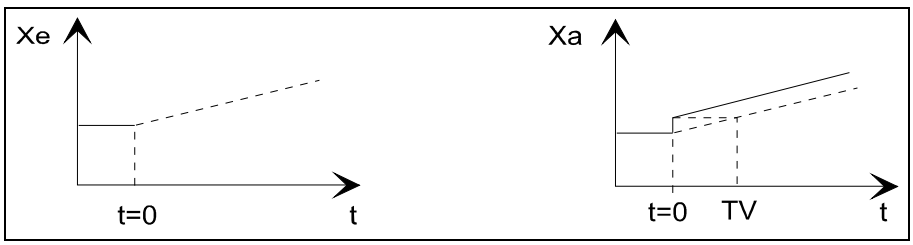
PI-Regler

Die Kenngrößen für einen PI-Regler sind der Proportionalbeiwert K und die Nachstellzeit TN. Der Proportionalbeiwert K ist der Quotient aus Ausgangsgröße und Eingangsgröße und bestimmt das P-Verhalten. Die Nachstellzeit TN ist diejenige Zeit, die bei der Antwort benötigt wird, um infolge der I-Wirkung eine gleichgroße Stellgrößenänderung zu erzielen, wie sie infolge des P-Anteils entsteht.



PD-Regler

Die Kenngröße für einen PD-Regler sind der Proportionalbeiwert K (siehe oben) und die Vorhaltezeit TV. Die Vorhaltezeit ist diejenige Zeit, die ein P-Regler bei konstanter Änderungsgeschwindigkeit der Eingangsgröße benötigen würde, um die gleiche Änderung der Ausgangsgröße zu bewirken, die ein PD-Regler infolge seines D-Anteils sofort bewirkt. Um die Vorhaltezeit TV zu bestimmen, geht man nicht von einer Sprungfunktion aus, sondern von einer linearen Änderung der Eingangsgröße.



PID-Regler

Die Kenngröße für einen PID-Regler sind der Proportionalbeiwert K, die Nachstellzeit TN und die Vorhaltezeit TV. Sie bestimmen jeweils das P-, I- und D-Verhalten.

Parameter-änderung

Der P-Anteil der Stellgröße wird nach folgender Formel gebildet:

$P\text{-Anteil} = KP \cdot (XW_k - XW_{k-1})$

Wird KP oder R im Automatikbetrieb geändert, wirkt sich dies nur auf nachfolgende Änderungen der Regeldifferenz XWk aus. Der momentane Wert der Stellgröße bleibt von der Parameteränderung unbeeinflusst. Dieses Verhalten ermöglicht eine stoßfreie Parameteränderung.

Ist dieses Verhalten jedoch nicht erwünscht, so kann es durch folgende Berechnung, die nur einmal bei jeder Parameteränderung vorzunehmen ist, beseitigt werden (Beispiel für KP-Änderung).

$Y_{k-1} = Y_{k-1} + XW_{k-1} (KP_{neu} - KP_{alt})$

Wird bei einer Parameteränderung folgendes Programm verwendet, verhält sich der Regler wie ein analoger Regler:

```
:L    KPneu    KPneuneu laden
:L    KPalt    KPalt laden
:-G
:L    DD38     XWk-1
:xG
:L    DD44     Yk-1
:+G
:T    DD44     =Yk-1
```

Abkürzungen für PID-Regler

dY_k	Berechnetes Stellinkrement
dZ_k	Störinkrement
GP	Gleitpunktdarstellung
k	k-te Abtastung
K	Proportionalbeiwert
LP	Linkspunktdarstellung
OG	Obere Grenze (Begrenzer)
R	R-Parameter
TA	Abtastzeit
TD	TV/TA
TI	TA/TN
t	Abtastzeitpunkt = k · TA
TN	Nachstellzeit
TV	Vorhaltzeit
UG	Untere Grenze (Begrenzer)
W_k	Sollwert
X_k	Istwert
XW_k	Regeldifferenz
Y_k	Berechnete Stellgröße
YA_k	Stellwert (Stellinkrement oder Stellgröße)
Z_k	Störgröße

Linkspunktzahl

Zur Darstellung einer Linkspunktzahl im Datenbaustein wird ein Wort benötigt. Die Zuordnung zwischen dezimal dargestellter Linkspunktzahl, dual dargestellter Linkspunktzahl und der Darstellung im Format KF am Programmiergerät ist im folgenden Beispiel dargestellt.

Linkspunktzahl in Dezimaldarstellung	Linkspunktzahl in Dualdarstellung	Festpunktzahl
-0,999...	1000000000000001	-32767
-0,75	1010000000000000	-24576
-0,5	1100000000000000	-16384
-0,25	1110000000000000	-8192
0	0000000000000000	0
+0,25	0010000000000000	+8192
+0,5	0100000000000000	+16384
+0,75	0110000000000000	+24576
+0,999...	0111111111111111	+32767

Negative Linkspunktzahlen ergeben sich in der Dualdarstellung durch Zweierkomplementbildung aus positiven Linkspunktzahlen.

Linkspunktzahlen (LP) lassen sich nach folgenden Beziehungen in die am Programmiergerät dargestellten Werte (KF) umrechnen:

$LP \cdot 32767 = KF$

mit $-1 < LP < +1$ und $-32767 \leq KF \leq +32767$.

OB 254/255: Einen Datenbaustein verschieben/duplizieren

Einleitung

Mit den Sonderfunktionen OB 254/255 übertragen Sie einzelne Datenbausteine vom Anwenderspeicher in das DB-RAM (Datenbausteinspeicher) der CPU. Die Sonderfunktionen OB 254 und OB 255 laufen identisch ab, wobei der OB 254 ausschließlich für DX-Datenbausteine und der OB 255 für DB-Datenbausteine zuständig ist.

Anwendung

Verschieben oder Duplizieren eines Datenbausteins.

Funktion

Verschieben

Ein Datenbaustein wird vom Anwenderspeicher in das DB-RAM verschoben.

Ein Datenbaustein im Anwenderspeicher wird unter Beibehaltung seiner ursprünglichen Bausteinnummer in das DB-RAM verschoben. Die neue Anfangsadresse wird in die Adressliste im DB 0 eingetragen.

Duplizieren

Ein Datenbaustein im Anwenderspeicher oder im DB-RAM wird mit einer neuen Baustein-Nr. in das DB-RAM kopiert. Die Anfangsadresse des neuen Datenbausteins wird in die Adressliste im DB 0 eingetragen. Die Anfangsadresse des alten Bausteins im DB 0 bleibt erhalten, d. h. der ursprüngliche Datenbaustein ist weiterhin gültig.

Die Eintragung der Anfangsadresse in den DB 0 erfolgt erst dann, wenn die Übertragung vollständig abgeschlossen ist und alle Kennungen im Bausteinkopf eingetragen sind. Der duplizierte Baustein wird also vom Systemprogramm erst nach der kompletten Übertragung als gültig bzw. vorhanden erkannt.

Hinweis!

Das Verschieben des DB 0 in das DB-RAM ist nicht möglich, da dieser sich bereits im DB-RAM befindet. Sie können den DB 0 jedoch duplizieren

Parameter

1. AKKU-1-L-L:

Nummer des zu kopierenden Bausteins,

zulässige Werte: 0 bis 255

(0 nur für DX oder Duplizieren eines DBs).

2. AKKU-1-H-L:

Mit dem Wert in AKKU-1-H-L legen Sie fest, ob Sie einen Baustein verschieben oder duplizieren wollen:

AKKU-1-H-L = 0:

Der Datenbaustein DB (Aufruf OB 255) oder DX (Aufruf OB 254) mit der in AKKU-1-L-angegebenen Nummer wird in das DB-RAM verschoben.

AKKU-1-H-L = Nr. für den neuen Baustein,

zulässige Werte: 1 bis 255

Der Datenbaustein DB (Aufruf OB 255) oder DX (Aufruf OB 254) mit der in AKKU-1-L-L angegebenen Nummer wird in das DB-RAM dupliziert und unter der in AKKU-1-H-L hinterlegten Nummer in den DB0 eingetragen.

Die Werte von AKKU-1-L-H und AKKU-1-H-H werden von den OB 254 und OB 255 nicht berücksichtigt und sind daher für die Parametrierung der OBs ohne Bedeutung.

Fehlerfälle

- Der zu verschiebende Datenbaustein ist nicht vorhanden (OB 19).
- Der Baustein ist bereits im DB-RAM vorhanden (OB 31)- (deshalb die Funktion nur einmal - vorzugsweise im Anlauf - ausführen).
- Der Speicherplatz im DB-RAM ist nicht ausreichend(OB 31).

Im Fehlerfall wird die Funktion nicht ausgeführt. Das Systemprogramm erkennt einen Laufzeitfehler und ruft den OB 19 oder OB 31 auf. Die weitere Fehlerreaktion hängt von der Programmierung des OB 19 bzw. OB 31 ab. Ist der OB 19 bzw. 31 nicht geladen, so geht die CPU in den Stoppzustand.

In beiden Fällen ist im AKKU-1-L eine Fehlerkennung hinterlegt, die den aufgetretenen Fehler näher erläutert.

Beispiel

Es wird angenommen, dass im Anwenderspeicher die Datenbausteine DB 3 und DB 4 definiert sind. Im DB-RAM soll außer DB 0 noch kein Datenbaustein DB vorhanden sein. Die nachfolgende Tabelle zeigt Ihnen welche Speicherbelegung sich nach mehrmaligen Aufruf des OB 255 mit den in der Tabelle aufgeführten Parametern ergibt.

Aufruf-Reihen-folge	Funktion	AKKU-1-				DB im Speicher nach Aufruf	
		-H-H	-H-L	-L-H	-L-L	Anwender-Speicher	DB-RAM
1	Verschieben	ohne	0	ohne	3	DB 4	DB 3
2	Duplizieren	Be-	5	Be-	4	DB 4	DB 3,5
3	Duplizieren	deu-	6	deu-	5	DB 4	DB 3,5,6
4	Verschieben	tung	0	tung	4	kein DB	DB 3,4,5,6

Teil 13 Integrierte Hantierungsbausteine

Überblick

In diesem Kapitel werden die Standardhantierungsbausteine (FB 120 - FB 127), sowie die Reglerbausteine (FB 250, FB 251) näher beschrieben. Sie werden eingesetzt, wenn Daten oder Parameter sowie Steuerungsinformationen von den Kommunikationsprozessoren übergeben werden sollen.

Die VIPA Standard-Funktionsbausteine (Best.-Nr.: VIPA-SW928) bilden ein fertiges Softwareprodukt.

Nachfolgend sind beschrieben:

- Parametrierung
- Anzeigenwort
- FB 120 bis FB 127
- FB 250, FB 251

Inhalt

Thema

Teil 13 Integrierte Hantierungsbausteine 13-1
Einführung 13-2
Die Parameter der Hantierungsbausteine 13-3
Parameterbeschreibung 13-4
Parametrierung von SSNR, A-Nr, ANZW und BLGR 13-6
Indirekte Parametrierung von Quell- und Zielangaben 13-9
Aufbau des Anzeigenwortes 13-10
Arbeiten mit dem Anzeigewort 13-12
Das Längen - Wort 13-15
Aufbau Parametrierfehlerbyte (PAFE) 13-16
Einstellbare Blockgrößen 13-16
Tabelle über die möglichen QTYP/ZTYP - Parameter 13-17
Der SEND - Baustein FB 120 13-19
Der RECEIVE-Baustein FB 121 13-22
Der FETCH-Baustein FB 122 13-25
Der CONTROL-Baustein FB 123 13-26
Der RESET-Baustein FB 124 13-27
Der SYCHRON-Baustein FB 125 13-28
SEND-ALL 126 13-29
RECEIVE-ALL 127 13-30
Blockgröße 13-31
Analogwert-Anpassungsbausteine FB 250 und FB 251 13-32
Beispiel für eine Analogwertverarbeitung 13-37

Einführung

Was sind Hantierungs-austeine?

Durch die mitgelieferten Hantierungsbausteine, wird der Einsatz von Kommunikationsprozessoren (z.B. CP 143, SSM-BG43D) und IPs im Automatisierungsgerät ermöglicht.

Hierdurch wird die Leistungsfähigkeit der CPU 928 erheblich gesteigert.

Die Hantierungsbausteine steuern den gesamten Datenaustausch zwischen der CPU und den CPs, IPs.

Vorteile der Hantierungsbausteine:

- Wenig Anwenderprogrammspeicherplatz geht verloren
- kurze Laufzeiten der Bausteine.

Die Hantierungsbausteine CPU 928 benötigen keine

- Merkerbereiche (Ausnahme RECEIVE-ALL =MW 252/254)
- Zeitbereiche
- Zählerbereiche.

Übersicht

Folgende Hantierungsbausteine stehen zur Verfügung:

FB 120	"SEND"
FB 121	"RECEIVE"
FB 122	"FETCH"
FB 123	"CONTRL"
FB 124	"RESET"
FB 125	"SYNCHRON"
FB 126	"SEND-ALL"
FB 127	"RECEIVE-ALL"
FB 250	"RLG:AE"
FB 251	"RLG:AA"

Die Parameter der Hantierungsbausteine

Übersicht

Alle nachfolgend behandelten Hantierungsbausteine haben eine einheitliche Schnittstelle zum Anwenderprogramm.

Die benutzten Parameter:

SSNR	:Schnittstellennummer
A-NR	:Auftragsnummer
ANZW	:Anzeigenwort (Doppelwort)
QTYP/ZTYP*	:Typ der Datenquelle bzw. des Datenziel
DBNR*	:Datenbausteinnummer
QANF/ZANF*	:Relative Anfangsadresse innerhalb des Typs
QLAE/ZLAE*	:Anzahl der Quell-/Zieldaten
PAFE	:Parametrierungsfehler
BLGR	:Blockgröße

*) Die bei einem bestimmten Aufruf nicht benötigten Parameter (z.B. bei "ALL"-Funktion), können bei der Parametrierung des Bausteins mit "CR" übersprungen werden.

Ein Hantierungsbaustein kann direkt oder indirekt parametriert werden. Nur der Parameter "PAFE" muss immer direkt angegeben werden. Bei der direkten Parametrierung verarbeitet der Hantierungsbaustein die beim Bausteinaufruf angegebenen Parameter unmittelbar. Bei der indirekten Parametrierung werden dem Hantierungsbaustein per Bausteinparameter Zeiger, die auf Parameterfelder (Datenbausteine/Datenworte) zeigen, übergeben.

Parameterbeschreibung

SSNR **Schnittstellen-Nummer**

Nummer der logischen Schnittstelle auf die sich der betreffende Auftrag bezieht.

Parameterart :Datum Byte (Format KY)

Sinnvoller Bereich :0,0...0,255 (bei direkter Parametrierung) *

*) Ist das linke Byte >0 handelt es sich um eine indirekte Parametrierung von SSNR, A-NR; ANZW bzw. BLGR.

Das rechte Byte stellt in diesem Fall den Zeiger auf dem Parameterfeld dar (siehe Beispiele: Indirekte Parametrierung)

A-NR **Auftragsnummer**

Angesprochene Auftragsnummer für die logische Schnittstelle

Parameterart :Datum Byte (Format KY)

Sinnvoller Bereich :0,0...0,223

Hierbei hat die Auftragsnummer 0 die Spezialfunktion "ALL", die beim FETCH Baustein nicht zugelassen ist

(siehe auch Beispiel ALL-Funktion).

ANZW **Anzeigenwort (Doppelwort)**

Adresse des Anzeigendoppelwortes im Anwenderspeicher, in dem die Abarbeitung des unter A-Nr. angegebenen Auftrages angezeigt wird.

Parameterart :Adresse Wort

Erlaubter Bereich :DW oder MW belegt wird jeweils DW und DW+1 oder MW und MW+2

Die Angabe DW bezieht sich auf den vor dem Aufruf aufgeschlagenen Datenbaustein.

Aufbau des ANZW: siehe Aufbau des Anzeigenwortes

QTYP/ZTYP **Typ der Datenquelle oder des Datenziels**

An diesem Parameter kann mittels ASCII-Zeichen der Typ, der Datenquelle (SEND) oder des Datenziels (RECEIVE und FETCH) angegeben werden.

Parameterart :Datum Zeichen (Format KC)

Sinnvoller Bereich :DB, MB, AB, EB, ZB, TB, AS, NN, XX, RW

DBNR **Datenbausteinnummer bei TYP XX, RW, DB**

Wurde bei QTYP/ZTYP die Kennung XX, RW oder DB angegeben, ist bei diesem Parameter die Nummer des gewünschten DBs mitzuteilen.

Parameterart :Datum Byte (Formt KY)

Sinnvoller Bereich :0,1...0,255.

QANF/ZANF **Anfangsadresse des Quell-/Ziel-Datenblocks**

Bei Typkennung XX und RW (indirekte Parametrierung) wird hier die DW-Nummer angegeben, ab der die Parameter liegen. Ansonsten bezieht sich dieser Parameter absolut auf den angegebenen Bereich.

Parameterart :Datum Festpunkt (Format KF)

Möglicher Bereich :0...+32767 bzw. bis -32768 bei "AS".

QLAE/ZLAE **Länge des Quell-/Ziel-Datenblocks**

Die Angabe wird je nach Quell-/Zieltyp in Bytes oder in Worten verstanden.

Parameterart :Datum Festpunktzahl (Format KF)

Möglicher Bereich :0...+32767 (-1)*

*) Die "Jokerlänge" (QLAE, ZLAE (-1)) bedeutet beim RECEIVE, dass so viele Daten übernommen werden, wie der Sender liefert oder bis eigene Bereichsgrenze erreicht wurde. Beim SEND, dass so viele Daten übergeben werden wie der Bereich zulässt

(Übertragung bis eigene Bereichsgrenze erreicht ist).

BLGR **Blockgröße**

Bei Neustart wird mit Hilfe des Hantierungsbausteins "SYNCHRON" die Blockgröße ausgehandelt (Größe der Datenblöcke die bei einem Verbindungaufbau zwischen AG zum CP ausgetauscht werden.)

Große Bockgröße = hoher Datendurchsatz und lange Laufzeit und damit hohe Zykluszeitbelastung.

Kleine Blockgröße = kleiner Datendurchsatz und kleine Laufzeiten

Parameterart :Datum Byte (Format KY)

Möglicher Bereich :0,0 ...0,255 (siehe Einstellbare Blockgrößen).

PAFE **Fehleranzeige bei Parametrierungsfehler**

Das hier angegebene "BYTE" (Ausgang, Merker) wird gesetzt, wenn der Baustein einen "Parametrierungsfehler" erkennt z.B. Schnittstelle (Anschaltung) nicht vorhanden oder unzulässige Parametrierung von QTYP/ZTYP; QANF/ZANF;QALAE/ZLAE erfolgte.

Parameterart :Ausgang, Byte

Sinnvoller Bereich :AB 0... AB 63 bei CPU 941; AB 0 ... AB127
bei CPU 942 :MB 0...MB 255

Der Aufbau des PAFE finden Sie in diesem Kapitel weiter hinten.

Parametrierung von SSNR, A-Nr, ANZW und BLGR

Das HIGH-Byte des Parameters SSNR dient als Umschaltkriterium für die direkte oder indirekte Parametrierung.

- HIGH-Byte von SSNR = 0 Bedeutet direkte Parametrierung
  SSNR, A-NR, ANZW oder BLGR sind direkt am Baustein vorgegeben.
- HIGH-Byte von SSNR ><0 Bedeutet indirekte Parametrierung
  SSNR, A-NR und ANZW/BLGR sind in dem aufgeschlagenen DB ab dem im LOW-Byte von SSNR angegebenen Datenwort abgelegt.

SSNR und A-NR haben in beiden Parametrierungsarten das gleiche Datenformat (KY). Beim Anzeigenwort ANZW unterscheiden sich die Darstellungsformate. Während bei der direkten Parametrierung die Adresse des Anzeigenwortes direkt (z.B. MW 100) angegeben wird, muss bei der indirekten Parametrierung eine zusätzliche Angabe über den Bereich des Anzeigenwortes erfolgen. Die Angabe dieses Bereiches steht in den, dem Anzeigenwort vorangestellten Datenwort in ASCII-Code verschlüsselt.

Hierbei steht: MW für Anzeigenwort im Merkerbereich
DB für Anzeigenwort im Datenbaustein.

In dem darauf folgenden Datenwort des Parameterbereiches im DB steht im Datenformat KY die ANZW-Adresse, bei DB zusätzlich die Bausteinnummer (im ersten Byte des KY Formates).

Beispiele

Direkte Parametrierung von SSNR, A-NR und ANZW (ANZW = MW)

```
          SPA      FB121
          NAME    :RECEIVE
          SSNR    :KY 0,3       ;Parameter SSNR =3
          A-NR    :KY 0,100     ;Parameter A-NR=Auftrag NR 100
          ANZW    :MW 240       ;Parameter ANZW = MW 240
    .
    .
 (ANZW = DW)

          A       DB            X
          .
          .
          SPA      FB123
          NAME    :CONTROL
          SSNR    :KY 0,3       ;Parameter SSNR = 3
          A-NR    :KY 0,100     ;Parameter A-NR = NR 100
          ANZW    :DW 40        ;Parameter ANZW =DW 40 im DB X
             .
             .



          DB      X

          DW 40
          DW 41
```

Indirekte Parametrierung von SSNR, A-NR und ANZW (Typ MW)

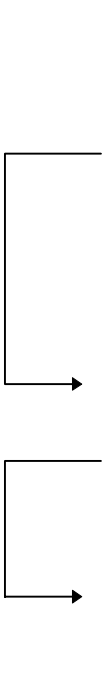
```
          A       DB X          ;Aufschlagen DB "X"
          SPA     FB 120
          NAME    :SEND
          SSNR    :KY 255,1     ;Kennung indirekte Param.
                                ;Wird als Zeiger interpretiert
          A-NR    :KY 0,0       ;Irrelevant
          ANZW    :MW 0         ;Irrelevant
             .
             .
          DB      X

          DW 1    :KY 0,1       ;Parameter SSNR (=1)
          DW 2    KY 0,31       ;Parameter A-NR (31)
          DW 3    KC MW        Parameterart für ANZW (=MW)
          DW 4    KY 0,200      ;Adresse des Parameters ANZW
                                ;(MW 200 und MW 202)
```

Fortsetzung ...

... Fortsetzung

Indirekte Parametrierung von SSNR, A-NR und ANZW (Typ DW)



```
    .
    .
A       DB X          ;Aufschlagen des DB "X"
SPA     FB 120
NAME    :SEND
SSNR    :KY 255,1     ;Kennung indirekte Param.
                      ;Wird als Zeiger interpretiert
A-NR    :KY 0,0       ;Irrelevant
ANZW    :MW 0         ;Irrelevant
    .
    .
DB      X

DW 1    KY 0,1        ;Parameter SSNR (=1)
DW 2    KY 0,31       ;Parameter A-NR (=31)
DW 3    KC DB         ;Parameterart für ANZW (=DW)
DW 4    KY 222,10     ;Adr. des ANZW
                      ;(im DB 222 DW10 + DW11)



DB 222
DW 10                 ;
DW 11                 ; ANZW
```

Indirekte Parametrierung von SSNR und BLGR (SYNCHRON)

```
A       DB X                    DB aufschlagen
SPA     FB 125
NAME    SYNCHR
SSNR    :KY 255, 100      ;Zeiger auf Parameterliste
BLGR    :KY 0,0           ;Irrelevant
    .
    :
DB      X

DW 100   KY 0,16          ;Parameter SSNR
DW 101   KY 0,6           ;BLGR
```

Indirekte Parametrierung von Quell- und Zielangaben

Parameter

Parameter bei indirekter Parametrierung XX und RW

a) bei "XX"

DB-NR wenn High-Byte = 0

QANF + 0 KC	QTYP/Ztyp jedoch nicht XX, RW
1 KY	DBNR DB-NR. bei "Typ" DB
2 KF	QANF/ZANF Anfangsadresse
3 KF	QLAE/ZLAE Länge

b) bei "RW" READ/WRITE)

DB-NR = DBNR wenn High-Byte = 0

QANF + 0 KC	QTYP jedoch nicht XX, RW	Beschreibung "Datenquelle"
1 KY	DBNR DB-NR. bei "Typ" DB	
2 KF	QANF Quell-Anfangsadresse	
3 KF	QLAE Quell-Länge	
4 KL	ZTYP jedoch nicht XX und WR	Beschreibung "Datenziel"
5 KY	DBNR DB-NR bei "ZTYP" DB	
6 KF	ZANF Ziel-Anfangsadresse	
7 KF	ZLAE Ziel-Länge	

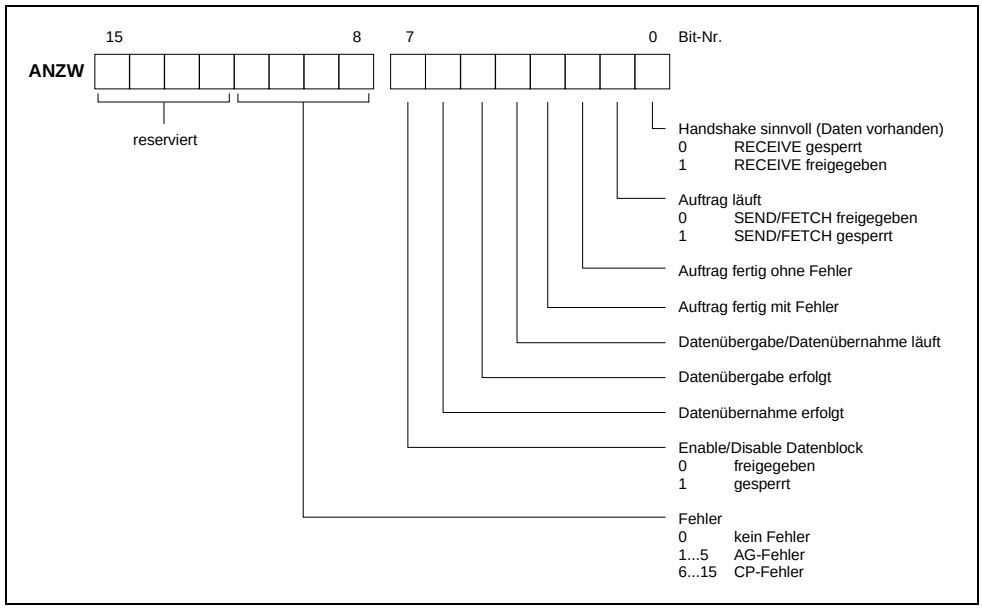
Bei Quell-/ Zieltyp "AS" (Absolutadresse sind im Datenformat KH auch Adressen bis FFFFh (65535) parametrierbar.

Aufbau des Anzeigenwortes

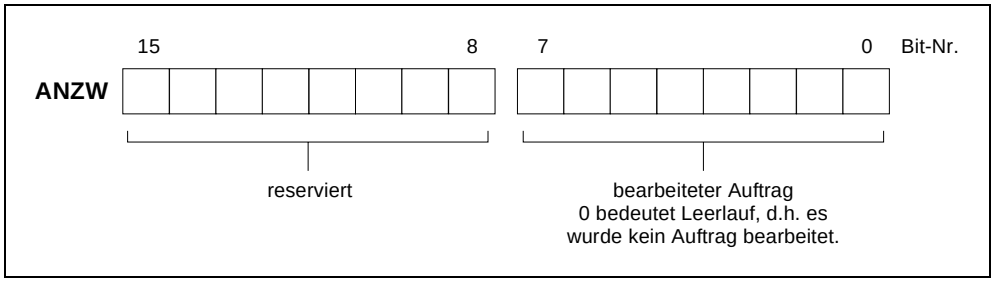
Funktion

Das "Anzeigenwort" zeigt den Zustand für einen bestimmten Auftrag auf einem CP an.

Im Programm sollte für jeden Auftrag ein eigenes "Anzeigenwort" für jeden definierten Auftrag bereitgestellt werden.



Im Anzeigenwort der "ALL"-Funktionen (Baustein-Anzeigenwort bei SEND-, RECEIVE ALL) tragen die Bausteine SEND- und RECEIVE die Auftragsnummer ein, für die sie im aktuellen Durchlauf aktiv waren.



Status und Fehleranzeige im Anzeigenwort

Das Anzeigenwort selbst ist in vier "Tetraden" aufgeteilt

- Tetrade 1 Bit 0 bis 3, Statusanzeigen des Auftrages:

- Hier ist verschlüsselt, ob ein Auftrag bereits gestartet ist. Ob ein Fehler aufgetreten ist, oder ob der Auftrag gesperrt ist, (z.B. weil die virtuelle Verbindung nicht besteht).

- Tetrade 2 Bit 4 bis 7 Datenverwaltung des Auftrages (Datenverkehr CPU - CP):

- Hier ist verschlüsselt, ob der Datentransfer für den Auftrag noch läuft, oder ob die Datenübergabe oder Datenübernahme bereits abgeschlossen ist.
- Mit dem Bit "Enable/Disable" kann der Datentransfer für den Auftrag gesperrt werden (Disable =1; Enable =0).

- Tetrade 3 Bit 8 bis 11 Fehleranzeigen des Auftrages:

- Hier werden die Fehleranzeigen des Auftrages angezeigt. Diese Fehleranzeigen sind nur gültig, wenn auch gleichzeitig das Bit "Auftrag fertig mit Fehler" in der Tetrade für den Status gesetzt ist.

Fehler

Folgende Fehleranzeigen sind möglich:

- 0 kein Fehler

  Ist das Bit "Auftrag fertig mit Fehler" trotzdem gesetzt, ist es ein Hinweis darauf, dass der CP den Auftrag nach einem NEUSTART oder RESET neu aufgebaut hat.
- 3 Bereich im AG zu klein oder nicht vorhanden

  6 bis F: Fehlermeldungen der CPs

  Die Fehlermeldungen 6 bis F werden von den CPs selbst erzeugt und sind daher als CP-spezifische Meldungen anzusehen.

  Abhilfe: Fehlerursache mittels der entsprechenden CP-Beschreibung suchen und beseitigen.

- Tetrade 4 Bit 12 bis 15 wird z.Z. nicht benützt.

Arbeiten mit dem Anzeigewort

Handshake sinnvoll

Bit 0 Handshake sinnvoll

Setzten/Löschen	Durch Hantierungsbausteine entsprechend der Anzeige im Auftragsstatus. Bit Handshake sinnvoll (=1) wird beim RECEIVE-Baustein genutzt (Telegramm vorhanden oder RECEIVE Anstoß möglich).
Auswerten	Durch den RECEIVE-Baustein; nur wenn Bit gesetzt, leitet der RECEIVE den Handshake mit dem CP ein. Durch den Anwender (Abfrage, ob Telegramm vorhanden).

Auftrag läuft

Bit 1 Auftrag läuft

Setzen	Durch Hantierungsbausteine, wenn Auftrag an CP erteilt.
Löschen	Durch die Hantierungsbausteine, wenn ein Auftrag vom CP abgearbeitet ist. (z.B. Quittung eingetroffen).
Auswerten	Durch die Hantierungsbausteine Wenn der "alte" Auftrag abgearbeitet wird, ein neuer Auftrag erteilt Durch den Anwender, um zu erfahren, ob das Triggern eines neuen Auftrags sinnvoll ist.

Auftrag fertig ohne Fehler

Bit 2 Auftrag fertig ohne Fehler

Setzten	Durch Hantierungsbaustein, wenn der entsprechende Auftrag ohne Fehler abgeschlossen wurde.
Löschen	Durch die Hantierungsbausteine, wenn der Auftrag erneut ausgelöst wird.
Auswerten	Durch den Anwender, um zu prüfen, ob der Auftrag fehlerlos abgeschlossen wurde.

Auftrag fertig mit Fehler

Bit 3 Auftrag fertig mit Fehler

Setzen	Durch Hantierungsbausteine, wenn der entsprechende Auftrag mit Fehler abgeschlossen wurde. Die Fehlerursache ist dann im Hight-Teil des Anzeigenwortes verschlüsselt.
Löschen	Durch den Hantierungsbaustein, wenn der Auftrag erneut ausgelöst wird.
Auswerten	Durch den Anwender, um zu prüfen, ob der Auftrag mit Fehler abgeschlossen wurde. Ist die Kennung "Auftrag fertig mit Fehler" gesetzt, steht im Hight-Byte des Anzeigenwortes die Fehlerursache.

Datenübergabe läuft

Bit 4 Datenübernahme / Datenübergabe läuft

Setzen	Durch die Hantierungsbausteine SEND,RECEIVE, wenn die Übergabe / Übernahme für einen Auftrag begonnen wurde, z.B. wenn Daten über die ALL-Funktion (DMA-Ersatz) ausgetauscht werden, der Anstoß jedoch mit SEND-Direkt erfolgte.
Löschen	Durch die Hantierungsbausteine SEND, RECEIVE, wenn der Datenaustausch für einen Auftrag beendet ist (letzter Teilblock übertragen).
Auswerten	Durch den Anwender. Während der Datenübertragung CP-AG darf der Anwender den Datensatz eines Auftrages nicht mehr verändern. Bei kleinen Datenpaketen ist dies unkritisch, da hierbei der Datenaustausch in einem Bausteindurchlauf erledigt werden kann. Größere Datenmengen können jedoch nur in Blöcken übertragen werden, wobei diese Blockung über mehrere AG-Zyklen verteilt wird. Um die Konsistenz der Daten zu wahren, muss der Anwender daher zuerst prüfen ob der Datenblock gerade übertragen wurde, bevor er die Daten eines Auftrags verändert.

Datenübergabe erfolgt

Bit 5 Datenübergabe erfolgt

Setzten	Durch den Hantierungsbaustein SEND, wenn die Datenübergabe für einen Auftrag erfolgt ist.
Löschen	Durch den Hantierungsbaustein SEND, wenn für einen neuen Auftrag (neue Triggerung) mit dem Transfer der Daten begonnen wurde. Durch den Anwender, wenn die Auswertung erfolgte (Flankenbildung).
Auswerten	Durch den Anwender. Mit diesem Bit kann der Anwender ermitteln, ob der Datensatz für einen Auftrag schon auf den CP übertragen wurde, und wann ein neuer Datensatz für einen laufenden Auftrag (z.B. zyklisch Übertragung) bereitgestellt werden kann.

Datenübernahme erfolgt

Bit 6 Datenübernahme erfolgt

Setzen	Durch den Hantierungsbaustein RECEIVE, wenn die Datenübernahme für einen Auftrag abgeschlossen wurde.
Löschen	Durch den Hantierungsbaustein RECEIVE, wenn für einen neuen Auftrag (neue Triggerung) mit dem Transfer der Daten ins AG begonnen wurde. Durch den Anwender, wenn die Auswertung erfolgte (Flankenbildung).
Auswerten	Durch den Anwender, mit diesem Bit kann der Anwender ermitteln, ob der Datensatz eines Auftrages schon auf das AG übertragen wurde und wann ein neuer Datensatz für einen laufenden Auftrag ins AG transferiert wurde.

Disable / Enable

Bit 7 Disable / Enable Datenblock

Setzen	Durch den Anwender, um das Beschreiben eines Bereichs durch den RECEIVE-Baustein oder das Auslesen aus einem Bereich durch den SEND-Baustein zu verhindern (nur beim 1. Datenblock)
Löschen	Durch den Anwender, um den zugehörigen Datenbereich freizugeben.
Auswerten	Durch die Hantierungsbausteine SEND, RECEIVE. Ist das Bit 7 gesetzt, führen die Bausteine keinen Datenverkehr durch, sondern melden dem CP den "Fehler".

Fehlerbyte

Bit 8 bis 11 Fehlerbyte

Übermittelt der CP eine Fehlerkennung für einen Auftrag (in den Auftragsstatus), tragen die Hantierungsbausteine diese Kennung in das Hight-Byte des Anzeigenwortes ein.

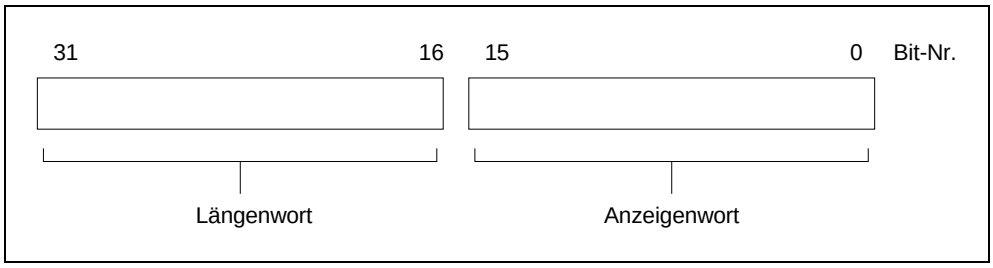
Das Längen - Wort

Beschreibung

Dem Anzeigenwort folgt unmittelbar d.h. adressenmäßig mit der nächsten Speicherzelle, das Längen-Wort.

Im Längenwort hinterlegen die Hantierungsbausteine (SEND, RECEIVE) die für den entsprechenden Auftrag bereits transferierten Daten: Die empfangene Daten bei Empfangsaufträgen und die bereits gesendete Daten bei Sendeaufträgen.

Die Anzeige im "Längenwort" erfolgt immer in Bytes und absolut.



Beschreiben Durch SEND, RECEIVE während des Datenaustausches.

Das Längen-Wort wird errechnet aus:

aktuelle Übertragungsanzahl + Anzahl bereits ausgetauschter Daten.

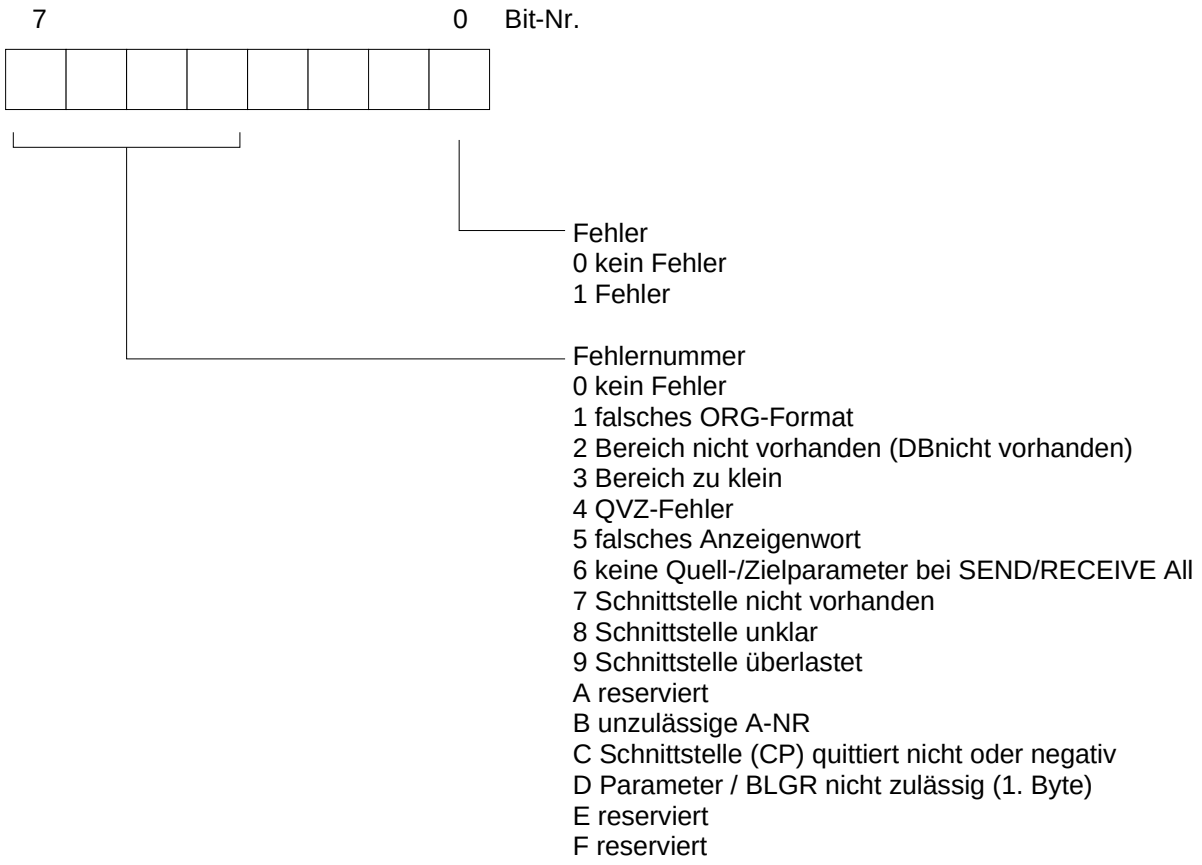
Löschen Durch Überschreiben oder mit jedem neuen SEND, RECEIVE, FETCH.

Auswerten Durch den Anwender

Wenn Bit "Auftrag fertig ohne Fehler" oder Datenübergabe / Übernahme erfolgt gesetzt ist, steht im Längen-Wort die aktuelle Quell- oder Ziellänge.

Wenn Bit "Auftrag fertig mit Fehler" gesetzt ist, beinhaltet das Längen-Wort die bis zum Fehlerfall übertragene Datenanzahl.

Aufbau Parametrierfehlerbyte (PAFE)



Einstellbare Blockgrößen

Als Blockgröße kann eingestellt werden:

0 Der Baustein benutzt die Default-Parameter
(Beim AG 115U auf 64 Byte eingestellt)
1 Blockung mit 16 Byte
2 Blockung mit 32 Byte
3 Blockung mit 64 Byte
4 Blockung mit 128 Byte
5 Blockung mit 256 Byte
6 Blockung mit 512 Byte
>6 und < 255 wie 0

Tabelle über die möglichen QTYP/ZTYP - Parameter

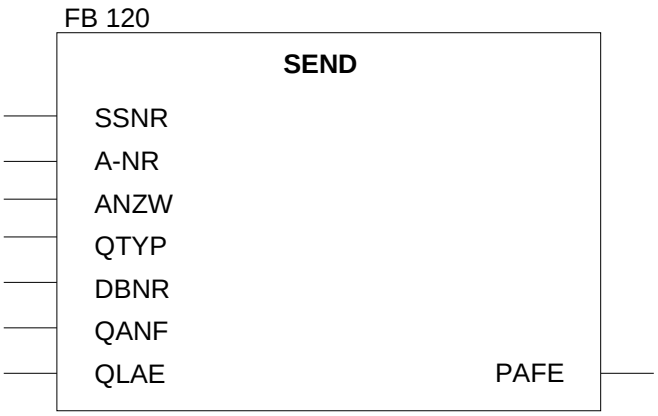
QTYP/ZTYP Beschreibung	NN	XX	RW	DB	MB	AB
	keine Quell-/Zielpara-meter am Baustein; Parameter müssen auf dem CP vorhanden sein	Indirekte Adressierung Parameter sind im (mit DBNR und QANF spez.) DB hinterlegt	Indirekte Adressie-rung ohne Daten-austausch; Quell-/ Ziel-parameter sind in einem DB hinterlegt	Quell-/Ziel-daten aus/im DB im Haupt-speicher	Quell-/Ziel-daten aus/im Merker-bereich	Quell-/Ziel-daten aus/im Prozess-abbild der Ausgänge (PAA)
DBNR Bedeutung	irrelevant	DB, in dem die Quell-/Zielpara-meter hinter-legt sind	DB in dem die Quell-/Zielpara-meter hinter-legt sind	DB, aus dem die Quelldaten entnommen werden oder in den die Zieldaten tranferiert werden	irrelevant	irrelevant
erlaubter Bereich		2...255	2...255	2...255		
QANF/ZANF Bedeutung	irrelevant	DW-Nr., ab der die Para-meter hinter-legt sind	DW-Nr., ab der die Parameter hinterlegt sind	DW-Nr., ab der die Daten entnommen oder eingeschrie-ben werden	Merkerbyte-Nr. ab der die Daten entnommen oder eingeschrie-ben werden	Ausgangs-byte-Nr., ab der die Daten entnommen oder eingeschrie-ben werden
erlaubter Bereich		0...2047	0....2047	0...2047	0...255	0...127
QLAE/ ZLAE Bedeutung	irrelevant	irrelevant	irrelevant	Länge des Quell-/Ziel-datenblocks in Worten	Länge des Quell-/Ziel-datenblocks in Bytes	Länge des Quell-/Ziel-datenblocks in Bytes
erlaubter Bereich				1...2048	1...255	1...128

... Fortsetzung

Fortsetzung ...

QTYP/ ZTYP Beschreibung	EB	PB	ZB	TB	AS
	Quell-Zieldaten aus /in Prozess-abbild der Eingänge(PAE)	Quell-Zieldaten aus /in Peri-pherie-bau-gruppen. Bei Quelldaten Ein-gabebaugr., bei Zieldaten Aus-gabebaugruppe	Quell-/Zieldaten aus /in Zählerzellen	Quell-/Zieldaten aus /in Zeitenzellen	Quell-/Zieldaten aus /in absolut adressierten Speicherzellen
DBNR Bedeutung	irrelevant	irrelevant	irrelevant	irrelevant	irrelevant
QANF/ZANF Bedeutung	Eingangsbyte-Nr. ab der die Daten entnom-men oder ein-geschrieben werden	Peripheriebyte-Nr. ab der die Daten entnom-men oder ein-geschrieben werden	Nummer der Zählerzelle, ab der die Daten entnommen oder einge-schrieben werden	Nummer der Zeitenzelle, ab der die Daten entnommen oder einge-schrieben werden	absolute An-fangsadresse, ab der die Da-ten entnommen oder einge-schrieben werden
erlaubter Bereich	0...127	0...127 digitale Peripherie 127...255 analoge Peripherie	0...127	0...127	0...+32767 0 -32768
QLAE/ZLAE Bedeutung	Länge des Quell-/Ziel-Datenblocks in Bytes	Länge des Quell-/Ziel-Datenblocks in Bytes	Länge des Quell-/Ziel-Datenblocks in Worten (Zählerzelle = 1 Wort)	Länge des Quell-/Ziel-Datenblocks in Worten (Zeitenzelle = 1 Wort)	Länge des Quell-/Ziel-datenblocks in Worten
erlaubter Bereich	1...128	1...256	1...128	1..128	1...32767

Der SEND - Baustein FB 120



Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	auszulösender Auftrag der Schnittstelle, d.h. Senden eines Telegramms starten
ANZW:	Adresse des Anzeigenworts (Doppelwort), in dem die Abarbeitung des ausgelösten Auftrages angezeigt wird.
QTYP:	Art der Datenquelle aus der die Daten übergeben werden
DBNR:	Nummer des Datenbausteins bei QTYP XX, RW (Read/Write), DB
QANF:	"relative Anfangsadresse" der Datenquelle
QLAE:	Anzahl der Quelldaten (in Bytes oder Worten)
PAFE:	Fehleranzeige bei Parametrierungsfehler

Funktions-beschreibung

Der SEND-Baustein dient zum Auslösen eines Auftrags zu einem CP mit oder ohne Datenübergabe.

SEND-ALL mit A-Nr.=0

Durch Aufruf des SEND mit A-Nr. = 0 wird ein SEND-ALL ausgelöst.

Die Beschreibung hierzu finden Sie auf der Folgeseite.

Alternativ steht Ihnen der FB 126 (SEND-ALL) zur Verfügung. Die Beschreibung hierzu finden Sie in diesem Kapitel weiter hinten.

Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	auszulösender Auftrag der Schnittstelle, d.h. Senden eines Telegramms starten. Für einen SEND-ALL-Auftrag ist hier 0 anzugeben.
ANZW:	Adresse des Anzeigenworts (Doppelwort), in dem die Abarbeitung des ausgelösten Auftrages angezeigt wird.
PAFE:	Fehleranzeige bei Parametrierungsfehler

Alle anderen Parameter sind bei diesem Auftrag irrelevant.

Die Adresse des Anzeigenwortes, die Angabe des Datentyps, der Anzahl und der Anfangsadresse der Daten muss bei der SEND-ALL -Funktion vom CP über den Kommunikationsbereich zur Verfügung gestellt werden.

Im Anzeigenwort, das dem betreffenden Auftrag zugeordnet ist, werden die Bits "ENABLE/DISABLE", "Datenübergabe erfolgt" sowie "Datenübergabe läuft" ausgewertet oder beeinflusst. Die Anzahl der übertragenen Daten zu einem Auftrag zeigt der SEND-ALL in dem ANZW folgenden Daten- bzw. Merkerwort an.

Anzeigenwort

Im Anzeigenwort, das am SEND-ALL-Baustein parametriert ist, wird die aktuelle Auftragsnummer hinterlegt (0 bedeutet Leerdurchlauf). Die Anzahl der übertragenen Daten zu einem Auftrag zeigt der SEND-ALL in dem Datenwort an, das dem Anzeigenwort folgt.

Der Aufruf der SEND-ALL-Funktion (d.h. der Aufruf des Send-Bausteins mit der ALL-Parametrierung) muss mindestens einmal im AG-Zyklus pro Schnittstelle erfolgen, wenn

- der CP selbständig Daten von einem AG anfordern kann; z.B. bei der CP 143 Plus mit der Auftragsart READ-Passiv.
- Ein CP-Auftrag mit einem SEND-DIREKT angestoßen wird, der CP die Daten zu diesem Auftrag jedoch erst über die "Hintergrundkommunikation" beim AG anfordert.
- Die Anzahl der Daten, die mit einem SEND-DIREKT dem CP übergeben werden sollen größer der eingestellten Blockgröße ist.

SEND-DIRECT

Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	auszulösender Auftrag der Schnittstelle, d.h. Senden eines Telegramms starten. Die Auftragsnummer darf nicht 0 sein.
ANZW:	Adresse des Anzeigenworts (Doppelwort), in dem die Abarbeitung des ausgelösten Auftrages angezeigt wird.
QTYP:	Art der Datenquelle aus der die Daten übergeben werden
DBNR:	Nummer des Datenbausteins bei QTYP XX, RW (Read/Write), DB
QANF:	"relative Anfangsadresse" der Datenquelle
QLAE:	Anzahl der Quelldaten (in Bytes oder Worten)
PAFE:	Fehleranzeige bei Parametrierungsfehler

Der "SEND-DIREKT"-Baustein wird im Normalfall im zyklischen Teil des Anwenderprogramms aufgerufen. Die Einbindung des Bausteins im Interrupt oder Weck-Programmteil ist zwar möglich, das Anzeigenwort (ANZW) kann hierbei jedoch nicht zyklisch aktualisiert werden (muss durch den CONTROL-Baustein übernommen werden).

Der Verbindungsaufbau mit dem CP wird für die Datenübergabe und für die Aktivierung eines Send-Anstoßes nur dann aufgenommen wenn:

- dem FB VKE "1" übergeben wurde und
- der CP den Auftrag freigegeben hat. (Bit "Auftrag läuft" im Anzeigewort =0).

Im Leerlauf des Bausteins wird nur das Anzeigenwort aktualisiert.

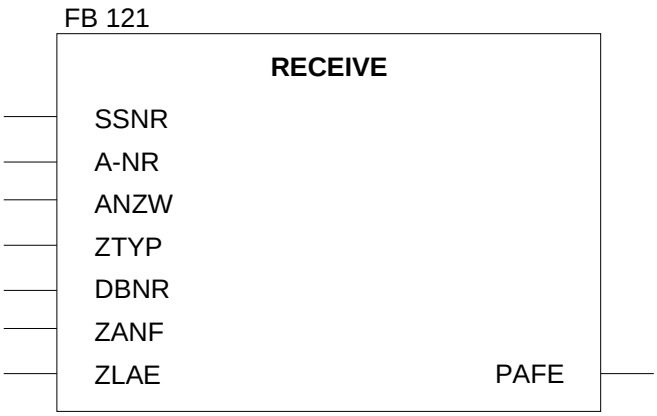
Ist im QTYP-Parameter die Kennung "NN" eingetragen, werden die Quellparameter des CP benutzt. Fehlen auch diese Parameter, wird der Auftrag mit einer Fehlermeldung beendet.

Kann der CP die Daten direkt übernehmen, überträgt der SEND-Baustein die geforderten Daten in einem Zug zum CP. Signalisiert der CP jedoch, dass er nur die Parameter des Auftrages wünscht oder die Anzahl der zu übergebenden Daten zu groß ist, werden dem CP nur die Parameter (QTYP/QLAE usw.), im zweiten Fall die Parameter mit dem ersten Datenblock übergeben. Die Daten oder der Folgeblock zu diesen Aufträgen fordert der CP über die SEND ALL-Funktion beim AG an. Hierzu ist es jedoch erforderlich, dass mindestens einmal im AG-Zyklus der SEND ALL Baustein aufgerufen wird. Die Bedienoberfläche ist in allen "Anstoßarten" für den Anwender der Bausteine gleich, nur der Zeitpunkt der Datenübergabe ist bei den zuletzt genannten Fällen um mindestens einen AG-Zyklus verschoben.

WRITE-Funktion

Die indirekt angegebenen Quell- und Zielparameter werden bei der Parametrierung mit QTYP = "RW" von "SEND" zum CP übertragen. Hier werden die Zielparameter zusammen mit den Nutzdaten (die über SEND ALL angefordert wurden) zum Kommunikationspartner gesendet.

Der RECEIVE-Baustein FB 121



Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	RECEIVE-Auftrag der Schnittstelle Abfrage nach einem Empfangstelegramm mit Datenübergabe
ANZW:	Adresse des Anzeigenwortes, in dem die Abarbeitung des Auftrags angezeigt wird
ZTYP:	Art des Datenziel, in das die Daten eingetragen werden sollen.
DBNR:	Nummer des DBs bei ZTYP XX, DB
ZANF:	relative Anfangsadresse des Datenziels
ZLAE:	Anzahl der Daten die übertragen werden sollen
PAFE:	Fehleranzeige bei Parametrierungsfehler

Funktions-beschreibung

Der RECEIVE-Baustein dient dem Abfragen oder dem Starten eines RECEIVE-Auftrages auf einem CP mit oder ohne Datenübernahme.

RECEIVE-ALL mit A-Nr.=0

Durch Aufruf des RECEIVE mit A-Nr. = 0 wird ein RECEIVE-ALL ausgelöst.

Die Beschreibung hierzu finden Sie weiter unten.

Alternativ steht Ihnen der FB 127 (RECEIVE-ALL) zur Verfügung. Die Beschreibung hierzu finden Sie in diesem Kapitel weiter hinten.

Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	RECEIVE-Auftrag der Schnittstelle Abfrage nach einem Empfangstelegramm mit Datenübergabe Für einen RECEIVE-ALL-Auftrag ist hier 0 anzugeben.
ANZW:	Adresse des Anzeigewortes, in dem die Abarbeitung des Auftrags angezeigt wird.
PAFE:	Fehleranzeige bei Parametrierungsfehler

Alle anderen Parameterangaben sind in dieser Betriebsweise irrelevant.

Die Adresse des Anzeigenworts, die Angabe des Typs, der Anfangsadresse und der Anzahl der Zieldaten stellt bei der RECEIVE ALL-Funktion der CP über den Kommunikationsbereich dem FB zur Verfügung.

Anzeigenwort

Im Anzeigenwort, das dem zu bearbeitenden Auftrag zugeordnet ist, werden die Bits "Enable/Disable", "Datenübernahme erfolgt" sowie "Datenübernahme/-übergabe läuft" ausgewertet oder beeinflusst und im Folgewort die "Empfangslänge" angezeigt.

Weiter wird die aktuelle Auftragsnummer hinterlegt, für den der RECEIVE ALL aktiv war. Im Leerlauf des RECEIVE ALL wird der Inhalt des Anzeigenworts gelöscht.

Der RECEIVE ALL ist immer dann mindestens einmal im AG-Zyklus aufzurufen, wenn

- der CP selbständig Daten an das AG abgeben will.
- Mit einem RECEIVE-DIREKT ein Datenblock empfangen werden soll, der größer als die eingestellte Blockgröße ist.

Der Funktionsbaustein "RECEIVE-ALL" kann vom Anwender aufgerufen werden im

- zyklischen Programmteil (z.B. im OB 1)
- zeitgesteuerten Programmteil (z.B Weck-Baustein)
- Interrupt-Programmteil (Prozessalarme)

RECEIVE-DIRECT Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	RECEIVE-Auftrag der Schnittstelle Abfrage nach einem Empfangstelegramm mit Datenübergabe
ANZW:	Adresse des Anzeigewortes, in dem die Abarbeitung des Auftrags angezeigt wird
ZTYP:	Art des Datenziel, in das die Daten eingetragen werden sollen.
DBNR:	Nummer des DBs bei ZTYP XX, DB
ZANF:	relative Anfangsadresse des Datenziels
ZLAE:	Anzahl der Daten die übertragen werden sollen
PAFE:	Fehleranzeige bei Parametrierungsfehler

Im Normalfall wird der RECEIVE-Baustein im zyklischen Teil des Anwenderprogramms aufgerufen. Die Einbindung des Bausteins im Interrupt oder Weck-Programmteil ist ebenso möglich, dabei wird jedoch das Anzeigenwort nicht zyklisch aktualisiert. Diese Funktion muss dann der CONTROL-Baustein übernehmen.

Der Quittungsverkehr mit dem CP (das heißt ein Auftragsanstoß) wird vom RECEIVE-Baustein in der "DIREKT"-Funktion nur aufgenommen wenn:

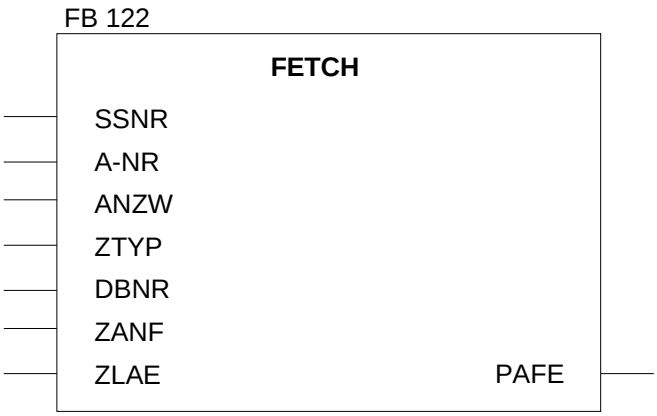
- dem FB VKE "1" übergeben wurde und
- der CP den Auftrag freigeben hat. (Bit "Handshake sinnvoll" = 1)

Im "Leerlauf" des Bausteins wird nur das Anzeigenwort aktualisiert.

Der RECEIVE-(DIRECT)-Baustein verhält sich unterschiedlich je nach Art der Versorgung und der CP-Reaktion:

- Ist im ZTYP-Parameter die Kennung "NN" eingetragen, erwartet der Baustein die Zielparameter (Typ, Anfang, Länge des Zieldatenblocks) vom CP. Wird vom CP ein Parametersatz geliefert, obwohl der RECEIVE-Baustein selbst mit den Zielparametern versorgt wurde, (ZTYP><NN), haben die Parameterangaben am Baustein Priorität gegenüber dem Parametersatz vom CP.
- Große Datenmengen können nur in Blöcken übernommen werden. Hierzu ist es notwendig, solche Folgeblöcke mit dem RECEIVE-ALL ins AG zu übertragen. Der Aufruf des RECEIVE-ALL mindestens einmal im AG-Zyklus (pro CP-Schnittstelle) ist daher immer dann notwendig, wenn mit einem CP größere Datenblöcke ausgetauscht werden sollen. Ebenso ist die zyklische Einbindung des RECEIVE-ALL notwendig, wenn der CP den RECEIVE-DIREKT nur zur Freigabe eines Empfangstelegramms benutzt und die Daten über die "Hintergrundkommunikation" dem AG übergibt.
- Die Parametrierung mit ZTYP = "RW" ist beim RECEIVE nicht erlaubt.

Der FETCH-Baustein FB 122



Parameter

SSNR:	Schnittstellennummer
A-NR:	auszulösender FETCH- oder READ-Auftrag
ANZW:	Anzeigenwort (Adresse des Anzeigenwortes)
ZTYP:	Art des Datenziels (DB, Merker, usw.)
DBNR:	Nummer des Datenbausteins bei ZTYP XX, RW, DB
ZANF:	relative Anfangsadresse des Datenziels
ZLAE:	Anzahl der Daten, die übernommen werden sollen
PAFE:	Fehleranzeige bei Parametrierungsfehler

Funktions-beschreibung

Der FETCH-Baustein dient dem Auslösen eines "Holauftrags" auf einem Kommunikationsprozessor. Die Empfangsdaten eines FETCH- oder READ-Auftrages können hierbei jedoch nur durch die RECEIVE ALL-Funktion ins AG übertragen werden. Der Baustein benötigt für die FETCH-Funktion alle Parameterangaben am Baustein; eine FETCH All-Funktion (Auftragsnr. = 0) ist nicht zugelassen.

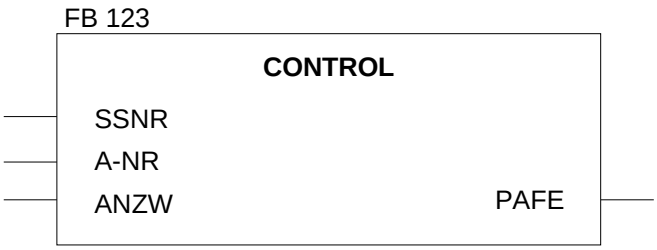
Die Zielparameter (ANZW, ZTYP, DBNR, ZANF, ZLAE) werden während des Quittungsverkehrs dem CP übergeben. Sobald die angeforderten Daten eingetroffen sind, stellt der CP diesen Parametersatz zusammen mit den Daten dem RECEIVE ALL-Baustein zur Verfügung. Der FETCH-Baustein selbst überträgt bzw. übernimmt keine Daten.

Bei der Parametrierung mit ZTYP = "RW" überträgt der FETCH die Quell- und Zielbeschreibung sowie die Adresse des Anzeigenworts zum CP.

Die FETCH-Funktion kann vom zyklischen, zeitgesteuerten oder interruptgesteuerten Programmteil aus aufgerufen werden. Die Übernahme der Daten erfolgt in jedem Fall durch einen RECEIVE ALL. Die Aktualisierung des Anzeigenwortes übernimmt der FETCH- oder CONTROL-Baustein.

Der Quittungsverkehr mit der Anschaltung für den Anstoß des FETCH wird nur aufgenommen, wenn dem Baustein VKE "1" übergeben und im entsprechenden CP-Anzeigenwort die Funktion freigeben wurde (Auftrag läuft = 0).

Der CONTROL-Baustein FB 123



Parameter

SSNR:	Schnittstellennummer
A-NR:	zu überwachender Auftrag des CPs z.B. Abfrage, ob der gestartete Auftrag mit oder ohne Fehler abgeschlossen wurde.
ANZW:	Anzeigenwort, in dem das Ergebnis der Abfrage dem Anwender mitgeteilt wird
PAFE:	Fehlerbyte bei Parametrierungsfehlern

Funktions-beschreibung

Der CONTROL-Baustein hat folgende Aufgaben:

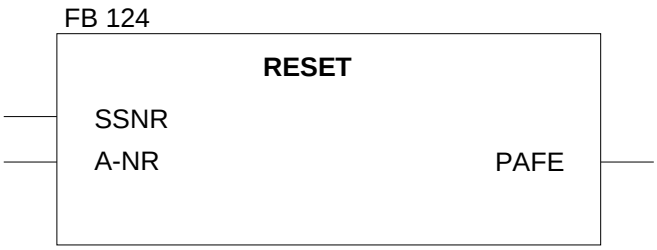
- Aktualisierung des Anzeigenworts nur dann wenn A-Nr. ≠ 0
- Abfrage, ob ein bestimmter Auftrag des CP zur Zeit "tätig" ist, z.B. Nachfrage nach einem Empfangstelegramm nur dann wenn A-Nr. ≠ 0
- Abfrage des CP, welcher Auftrag z.Zt. bearbeitet wird nur dann wenn A-Nr. = 0

Der CONTROL-Baustein nimmt keinen Quittungsverkehr mit dem CP auf, sondern überträgt nur die Anzeigen aus dem "Auftragsstatus" zum parametrierten Anzeigenwort. Der Baustein ist nicht VKE abhängig und sollte im zyklischen Teil des Programms aufgerufen werden.

Bei der CONTROL-DIRECT-Funktion (Parameter A-NR ≠ s0) wird das Anzeigenwort in der gleichen Weise aufgebaut und bearbeitet wie bei allen anderen "Hantierungsbausteinen (siehe Beschreibung des Baustein-parameters ANZW).

Wird der Parameter "Auftragsnummer" mit 0 versorgt, überträgt der CONTROL-Befehl den Inhalt der Auftragsstatuszelle 0 zum LOW-Teil des Anzeigenworts. In die Auftragsstatuszelle 0 schreibt der CP die Nummer des aktuellen Auftrags (das heißt des Auftrags, der gerade bearbeitet wird, wie z.B. die Auftragsnummer eines Telegramms).

Der RESET-Baustein FB 124



Parameter

SSNR:	Schnittstellennummer
A-NR:	Nummer des rückzusetzenden Auftrages
PAFE:	Fehlerbyte bei Parametrierungsfehler

Betriebsarten

Der Baustein kennt folgende beiden Betriebsarten

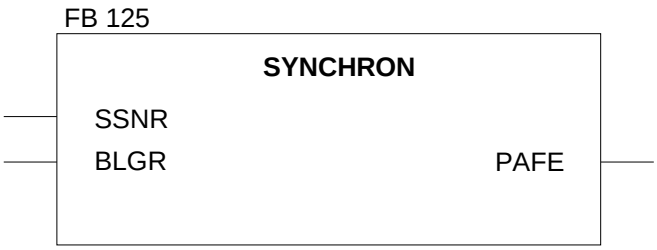
- RESET ALL
- RESET DIREKT

Funktions-beschreibung

Die RESET ALL-Funktion wird mit der Auftragsnummer 0 angewählt. Sie setzt alle Aufträge dieser logischen Schnittstelle zurück z.B. löscht sie alle Auftragsdaten und bricht alle laufenden Aufträge ab. Mit einer "direkten" Funktion (A-Nr. $\neq$ 0) wird nur der angegebene Auftrag auf der logischen Schnittstelle rückgesetzt.

Der Baustein arbeitet VKE-abhängig und kann von zyklischen, zeitgesteuerten oder alarmgesteuerten Programmteilen aus aufgerufen werden.

Der SYCHRON-Baustein FB 125



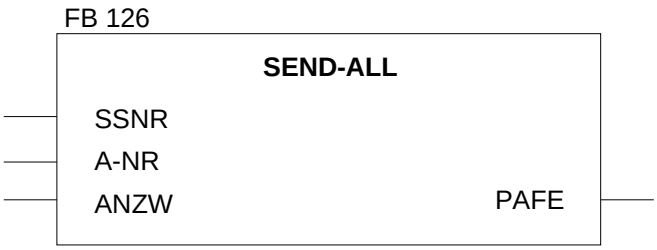
Parameter

SSNR:	Schnittstellennummer
BLGR:	Blockgröße
PAFE:	Parametrierungsfehler

Funktions-beschreibung

Der Baustein stellt im AG-Anlauf die Synchronisation zwischen AG und CP her und ist daher in den Anlauf-OBs aufzurufen. Gleichzeitig wird der Übergabebereich der Schnittstelle gelöscht und voreingestellt, sowie die Blockgröße zwischen CP und AG ausgehandelt.

SEND-ALL 126



Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	auszulösender Auftrag der Schnittstelle. Bei SEND-ALL muss A-NR immer 0 sein
ANZW:	Adresse des Anzeigenworts (Doppelwort), in dem die Auftragsnummer des aktuellen Auftrags steht
PAFE:	Fehleranzeige bei Parametrierungsfehler

Funktions-beschreibung

Als Auftragsnummer ist immer 0 anzugeben.

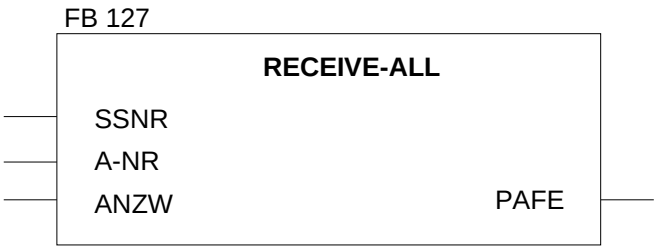
Im Anzeigenwort wird die Aktuelle Auftragsnummer hinterlegt (0 bedeutet Leerdurchlauf). Die Anzahl der übertragenen Daten zu diesem Auftrag zeigt der SEND-ALL in dem Merker-/Datenwort an, das dem Anzeigenwort folgt.

Der Aufruf des SEND-ALL muss mindestens einmal im AG-Zyklus pro Schnittstelle erfolgen.

Beispiel

```
00014      :
00016      :SPA FB 126
      NAME #SEND-A
      SSNR       =KY 0,0
      A-NR       =KY 0,0
      ANZW       =MW 120
      PAFE       =MB 198
00022      :
```

RECEIVE-ALL 127



Parameter

SSNR:	Schnittstellennummer, Nummer der logischen Schnittstelle
A-NR:	auszulösender Auftrag der Schnittstelle
ANZW:	Adresse des Anzeigenworts (Doppelwort), in dem die Auftragsnummer des aktuellen Auftrags steht
PAFE:	Fehleranzeige bei Parametrierungsfehler

Funktions-beschreibung

Als Auftragsnummer ist immer 0 anzugeben.

Im Anzeigenwort wird die Aktuelle Auftragsnummer hinterlegt (0 bedeutet Leerdurchlauf). Die Anzahl der übertragenen Daten zu diesem Auftrag zeigt der SEND-ALL in dem Merker-/Datenwort an, das dem Anzeigenwort folgt.

Der Aufruf des RECEIVE-ALL muss mindestens einmal im AG-Zyklus pro Schnittstelle erfolgen.

Beispiel

```
00024      :SPA FB 127
      NAME #REC-A
      SSNR       =KY 0,0
      A-NR       =KY 0,0
      ANZW       =MW 124
      PAFE       =MB 197
```

Blockgröße

Blockgröße Als Blockgröße kann eingestellt werden:

Kennung	Blockgröße
0	64 Byte (Standardwert)
1	16 Byte
2	32 Byte
3	64 Byte
4	128 Byte
5	256 Byte
6	512 Byte
7...254	Standardwert wie 0
255	512 Byte

Analogwert-Anpassungsbausteine FB 250 und FB 251

Einleitung

Diese Bausteine nehmen Umrechnungen vor, zwischen dem Nennbereich einer Analogbaugruppe und einem normierten Bereich, den der Anwender bestimmen kann.

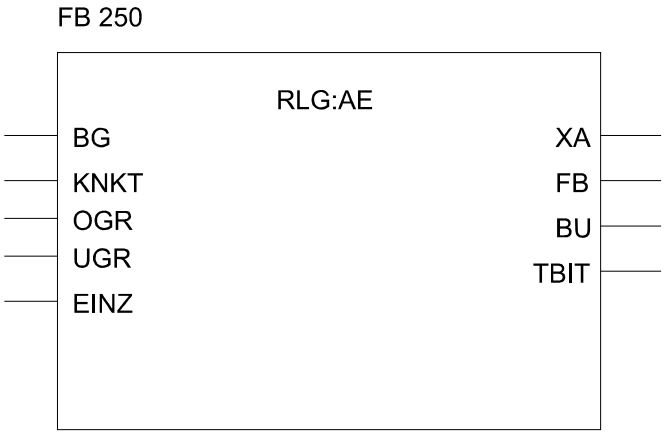
Die Parametrierung der Analogmodule muss mit dem Kanaltyp übereinstimmmen! Die Angabe wird nicht überprüft.

Analowert einlesen und normieren FB 250

Dieser Funktionsbaustein liest einen Analogwert einer Analog-Eingabe-baugruppe und liefert am Ausgang einen Wert XA in einem von Ihnen festgelegten (normierten) Bereich.

Mit den Parametern "Obergrenze (OGR)" und "Untergrenze (UGR)" legen Sie den gewünschten Bereich fest.

Die Art der Analogwertdarstellung der Baugruppe (Kanaltyp) muss im Parameter KNKT angegeben werden. Überschreitet der Analogwert den Nennbereich, wird der Parameter BU gesetzt.



Aufruf und Parameter

Parameter	Bedeutung	Art	Typ	Belegung	AWL
BG	Baugruppen-adresse	D	KF	128...224	: SPA FB 250 NAME: RLG:AE
KNKT	KN=Kanal-nummer KT=Kanaltyp	D	KY	KY=x,y x=0...15 y=3...6 3:Betragdarst. (4...20mA) 4:unipolare Darstellung 5:Betragszahl bipolar 6:Festpunktzahl bipolar (Zweierkompl.)	BG : KNKT: OGR : UGR : EINZ: XA : FB : BU : TBIT:
OGR	Obergr. des Ausgangs-wertes	D	KF	-32768... +32767	
UGR	Untergr. des Ausgangs-wertes	D	KF	-32768... +32767	
EINZ	Einzelab-tastung	E	BI	Bei "1" wird eine Einzelabtastung angeregt	
XA	Ausgangswert	A	W	normierter Analog-wert ist "0" bei Drahtbruch	
FB	Fehlerbit	A	BI	Ist "1" bei ungült. Kanal- oder Steckplatznr., bei ungültigem Kanal-typ und QVZ der Baugruppe	
BU	Bereichsüber-schreitung	A	BI	Ist "1" bei Über-schreitung des Nennbereichs	
TBIT	Tätigkeitsbit des FBs	A	BI	bei Signalzustand "1" führt der FB gerade eine Einzelabtastung durch	

KF = fester Wert

KY = Wort

Normierungs-schema

Der Funktionsbaustein FB 250 rechnet den gelesenen Wert linear auf die angegebene obere und untere Grenze (OGR und UGR) um, und zwar nach folgenden Formeln:

für Kanaltyp 3 (Betragszahl 4 bis 20 mA):

$$XA = \frac{UGR \cdot (2560 - xe) + OGR \cdot (xe - 512)}{2048}$$

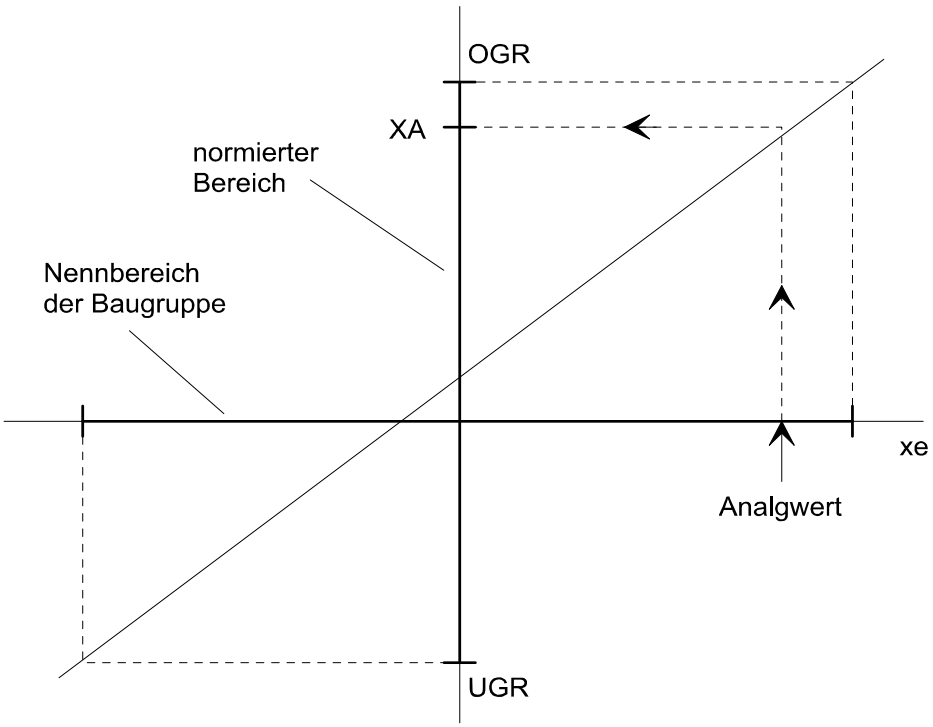
für Kanaltyp 4 (unipolare Darstellung):

$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot xe)}{2048}$$

Kanaltyp 5 und 6 (bipolare Darstellung):

$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot (xe + 2048)}{4096}$$

dabei bedeutet : XA vom FB ausgegebener Wert
xe von der Baugruppe gelesener Analogwert



Einzelabtastung

Der FB 250 gestattet das Lesen eines Analogwerts mit Einzelabtastung.

Setzt man den Parameter EINZ auf "1", so wird die Analogeingabebaugruppe veranlasst, den Analogwert des angewählten Kanals sofort in einen digitalen Wert umzuwandeln.

Während der Umrechnung (ca. 60ms) darf keine weitere Einzelabtastung angestoßen werden, die auf diese Baugruppe zugreift. Deshalb setzt der gerade aktive FB das TBIT so lange auf "1", bis der umgewandelte Wert eingelesen ist. Nach Beendigung der Einzelabtastung wird das TBIT wieder auf "0" gesetzt.

Analogwert ausgeben FB 251

Normierung

Mit diesem Funktionsbaustein lassen sich Analogwerte an Analog-Ausgabebaugruppen ausgeben. Die Art der Analogwertdarstellung der Baugruppe (Kanaltyp) muss im Parameter KNKT angegeben werden. Dabei werden Werte aus dem Bereich zwischen den Parametern "Untergrenze (UGR)" und "Obergrenze (OGR)" auf den Nennbereich der jeweiligen Baugruppe umgerechnet, und zwar nach folgenden Formeln:

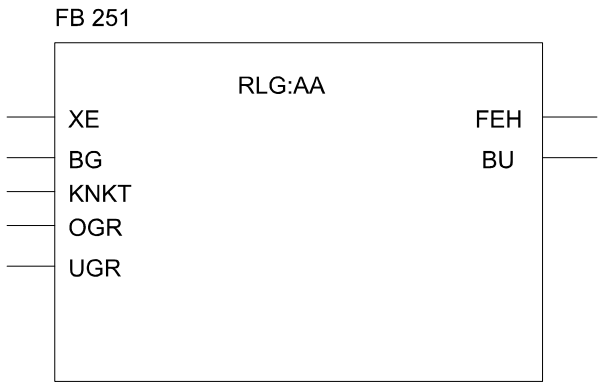
für Kanaltyp 0 (unipolare Darstellung):

$$xa = \frac{1024 \cdot (XE - UGR)}{OGR - UGR}$$

für Kanaltyp 1 (bipolare Darstellung):

$$xa = \frac{1024 \cdot (2 \cdot XE - OGR - UGR)}{OGR - UGR}$$

dabei bedeutet: XE am Funktionsbaustein angebener Digitalwert
xa zur Baugruppe geschriebener Wert



Aufruf und Parametrierung

Parameter	Bedeutung	Art	Typ	Belegung	AWL
XE	auszugebender Analogwert	E	W	Eingangswert (Festpunkt) im Bereich UGR...OGR	
BG	Baugruppen-adresse	D	KF	128...240	: SPA FB 250 NAME: RLG:AE
KNKT	KN=Kanal-nummer KT=Kanaltyp	D	KY	KY=x,y x=0...7 y=0;1;2 0: unipolare Darstellung 1:Festpunktzahl bipolar 2: Betragsdarst. (4...20mA bzw.1,5V)	BG : KNKT: OGR : UGR : EINZ: XA : FB : BU : TBIT:
OGR	Obergr. des Ausgangs-wertes	D	KF	-32768... +32767	
UGR	Untergr. des Ausgangs-wertes	D	KF	-32768... +32767	
FEH	Fehler bei der Grenzwert-vorgabe	A	BI	Ist "1",wenn UGR=OGR, bei ungültiger Kanal- oder Steckplatznr. bei ungültigem Kanaltyp und bei QVZ der Baugruppe	
BU	anzugebender Analogwert überschreitet UGR oder OGR	A	BI	Bei "1" liegt XE außerhalb (UGR;OGR) XE nimmt den Grenzwert an	

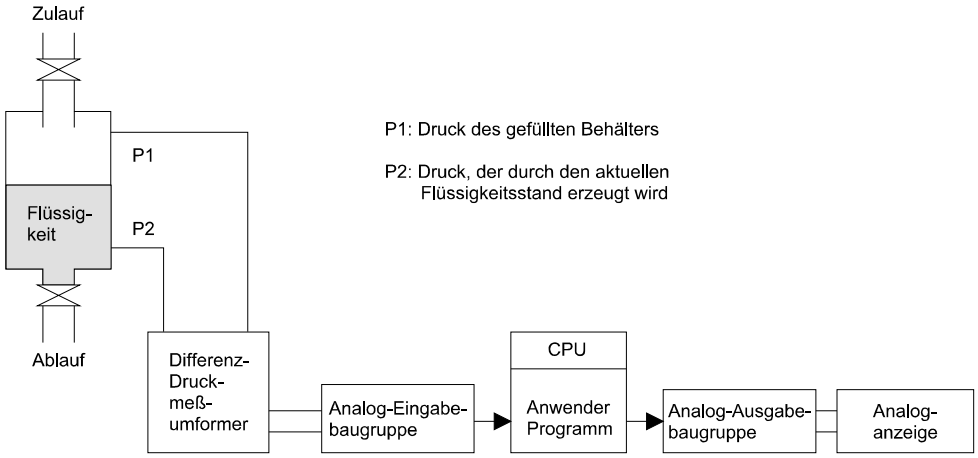
Beispiel für eine Analogwertverarbeitung

Aufgabenstellung

In einem geschlossenen Behälter befindet sich eine Flüssigkeit. Die aktuelle Niveauhöhe soll auf einem Anzeigeinstument jederzeit abgelesen werden können. Außerdem soll bei Erreichen eines vorgegebenen Grenzwertes eine Meldung abgegeben werden.

Beschreibung

- Die Füllstandshöhe (zwischen 0 und 10m) wird von einem Messumformer 0 - 20mA an eine Analogeingabebaugruppe weitergegeben.
- Die Analogeingabebaugruppe wandelt die analogen Stromwerte in digitale Einheiten (0 - 2048 Einheiten) um, die vom Anwenderprogramm des AG 115U weiterbearbeitet werden können.
- Das Anwenderprogramm prüft die eingelesenen Werte auf einen Grenzwert (max. zulässige Füllstandshöhe), gibt gegebenenfalls eine Meldung aus und übermittelt diese Werte an eine Analogausgabebaugruppe.
- Die Analogausgabebaugruppe setzt die Werte wieder in Spannungen (0 - 10V) um. Die Analoganzeige reagiert auf diese Spannungen mit einem der Füllstandshöhe proportionalen Zeigerausschlag.



Programmstruktur

- Funktionsbaustein FB 250 "Analogwert einlesen" aufrufen und parametrieren (Umwandlung des Werts in einem Bereich zwischen 0 und 1000cm [XA-Parameter]).
- Grenzwert bilden (PB9).

  Ein Überschreiten des Flüssigkeitsniveaus von 900cm verursacht eine Meldung (M12.6).
- Funktionsbaustein FB251 "Analogwert ausgeben" aufrufen und parametrieren (Umwandlung des zwischen 0 und 1000 cm liegenden Wertes [XE-Parameter] in einen Wert zwischen 0 und 1024 Einheiten für die Analog-Ausgabebaugruppe).

Beispiel

PB 1

```
          :SPA FB 250
NAME      :RLG:AE
BG        :KF     +128   Baugruppen-Anfangsadresse : 128
                         (bei fester Steckplatzadressierung
                         Steckplatz 0)
KNKT      :KY 0,4        Kanalnummer:0;unipolare Darstellung:4
OGR       :KF +1000      Physikalischer Messbereich:
UGR       :KF +0         0<XA<1000 cm
EINZ      :M 12.0        nur bei Einzelabtastung relevant:
                         (Einst. im Beispiel: Zykl. Bearb.)
XA        :MW 10         im MW 10: XA-Wert 0<XA<1000cm
FB        :M 12.1        relevant nur, wenn Drahtbruch
                         eingestellt ist
BU        :M 12.2        wenn Höhe > 1000cm, BU = 1.
TBIT      :M 12.3        nur bei Einzelabtastung relevant.
          :
          :SPA PB 9      Grenzwert bilden
          :
          :SPA FB 251    Analogwert ausgeben
NAME      :RLG:AA
XE        :MW 10         XA (FB 250) = XE (FB 251)
BG        :KF +160       Baugruppen-Anfangs-Adresse: 160
                         (bei fester Steckplatzadressierung
                         Steckplatz 1)
KNKT      :KY 0,0        Kanalnummer: 0;
                         unipolare Darstellung: 0
OGR       :KF +1000      Physikalischer Messbereich:
UGR       :KF +0         0<XA<1000cm
FEH       :M 12.4        wenn UGR = OGR, FEH = 1
BU        :M 12.5        wenn XA<UGR oder XA>OGR, BU = 1.
          :BE
```

PB 9

```
          :L KF 900      Niveau Maximalwert
          :L MW 10       Messwert
          :<=F           Vergleiche, ob Messwert > 900 Ist
          :=M 12.6       wenn ja, M12.6 = Reaktion noch im
                         gleichen Programmzyklus auslesen.
          :BE
```

Teil 14 Rechnerkopplung RK512

Überblick

Die zweite Schnittstelle SI 2 der CPU 928 können Sie nutzen als Schnittstelle für die Standard-Rechnerkopplung RK 512 mit der Prozedur 3964 oder 3964R.

Die Rechnerkopplung RK512 der CPU 928 verwendet den identischen Telegrammaufbau wie die Kommunikationsprozessoren SSM BG4xD.

Nachfolgend sind beschrieben:

- Grundlagen und Anwendungsbereiche von RK512
- Parametrierung
- Inbetriebnahme und Betrieb
- Telegrammbeschreibung

Inhalt

Thema	Seite
Teil 14 Rechnerkopplung RK512	**14-1**
Einführung RK512	14-2
Parametrierung	14-10
Koordinierungsmerker (KM)	14-26
Inbetriebnahme	14-28
Betrieb	14-30
Datenaustausch	14-38
Telegrammbeschreibung	14-43

Einführung RK512

Mit der Rechnerkopplung RK512 können zwischen zwei Punkt-zu-Punkt gekoppelten Kommunikationspartnern Daten ausgetauscht werden.

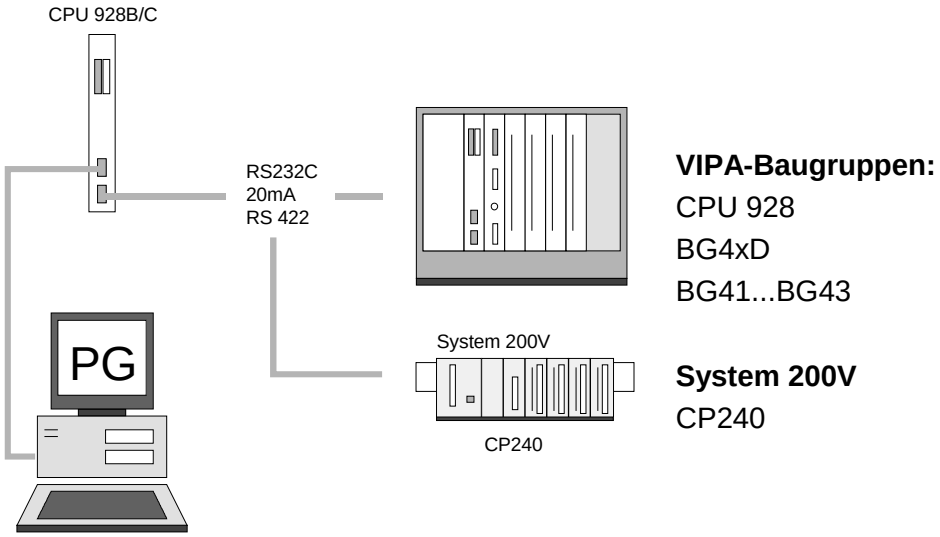
Die Rechnerkopplung RK512 gewährleistet größtmögliche Übertragungssicherheit, da das Protokoll der RK512 mehrere Schichten des ISO/OSI-Schichtenmodells (ISO IS 7498) umfasst:

- Bit-Übertragungsschicht (Schicht 1):
  In dieser Schicht wird die physikalische Übertragung der Datenbytes festgelegt (Physik, Baudrate;...).
- Sicherungsschicht (Schicht 2):
  Die Datenbytes werden mit der Übertragungsprozedur 3964(R) übertragen. Diese fügt den Datenbytes Start- und Endezeichen hinzu und führt bei Fehlern eventuelle Wiederholungen durch.
- Vermittlungsschicht (Schicht 3):
  Ist bei der RK 512 nicht vorhanden, da es sich bei ihr um eine reine Punkt-zu-Punkt-Verbindung handelt.
- Transportschicht (Schicht4):
  Die RK 512 beantwortet jedes korrekt empfangene Befehlstelegramm mit einem Reaktionstelegramm. Damit kann der Absender überprüfen, ob seine Daten vollständig beim Partner angekommen sind bzw. ob seine angeforderten Daten beim Partner verfügbar sind.

Anwendungs-bereich

Die Rechnerkopplung RK512 realisiert den direkten Datenaustausch zwischen der CPU 928 in einem AG-135U bzw. AG-155U und einem der folgenden Partner:

- eine weitere CPU 928
  in dem AG-135U und 155U
- BG41, BG42, BG43
  SSM-Baugruppe mit Kommunikationsprozessor in dem AG-115U, 115H, 135U, 155U und 155H
- BG4xD
  CP-Baugruppe in den AG-115U, 115H, 135U, 150U, 155U und 155H
- CP240
  CP-Modul des System 200V von VIPA
- ein Fremdrechner und Baugruppen, die die RK512 beherrschen.



Für die Rechnerkopplung RK512 stehen Ihnen verschiedene Schnittstellen-hybridmodule zur Verfügung, die Sie bei VIPA bestellen können:

- 20mA-Modul
- RS 232C-Modul
- RS 422-Modul

Die Verbindung zum Partner stellen Sie mit einer Standard-Steckleitung her. Die Übertragungsgeschwindigkeit für den Datenaustausch stellen Sie entsprechend dem jeweiligen Koppelpartner ein.

Zugelassen sind:

- 110 Bit/s bis 38 400 Bit/s bei RS422-Übertragung
- 110 Bit/s bis 38 400 Bit/s bei RS232C-Übertragung
- 110 Bit/s bis 38 400 Bit/s bei 20mA-Übertragung

Aufträge

Mit der Standard-Rechnerkopplung RK512 können Sie **aktiv**:

- Daten senden: **SEND**-Auftrag
- Daten holen: **FETCH**-Auftrag

Mit der Standard-Rechnerkopplung RK512 können Sie **passiv**:

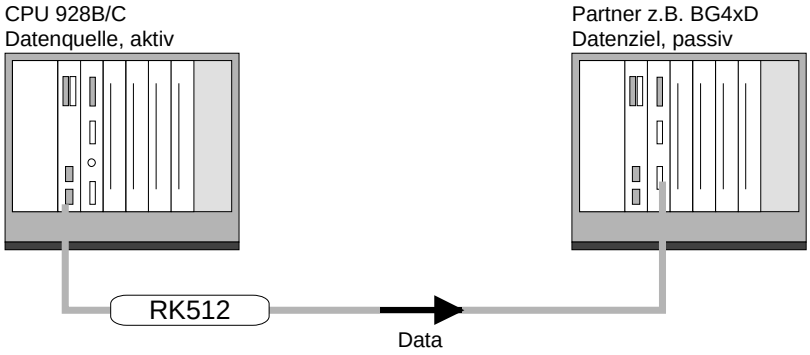
- Daten empfangen: Partner-**SEND**-Auftrag
- Daten zum Abholen bereitstellen: Partner-**FETCH**-Auftrag

SEND-Auftrag und Partner-SEND-Auftrag bilden ein Auftragspaar. Ein SEND-Auftrag bei der CPU 928 führt zu einem Partner-SEND-Auftrag beim Partner und umgekehrt.

FETCH-Auftrag und Partner-FETCH-Auftrag bilden ein Auftragspaar. Ein FETCH-Auftrag bei der CPU 928 führt zu einem Partner-FETCH-Auftrag beim Partner und umgekehrt.

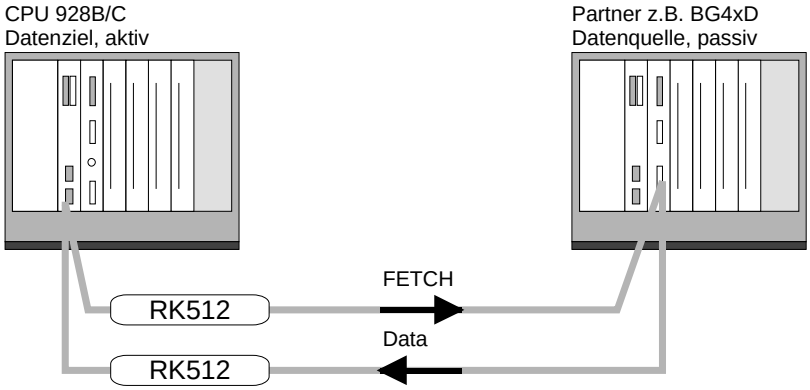
SEND-Auftrag

Die CPU 928 ist aktiv, d.h. sie sendet Daten zum Partner. Der Partner ist passiv - er empfängt Daten.



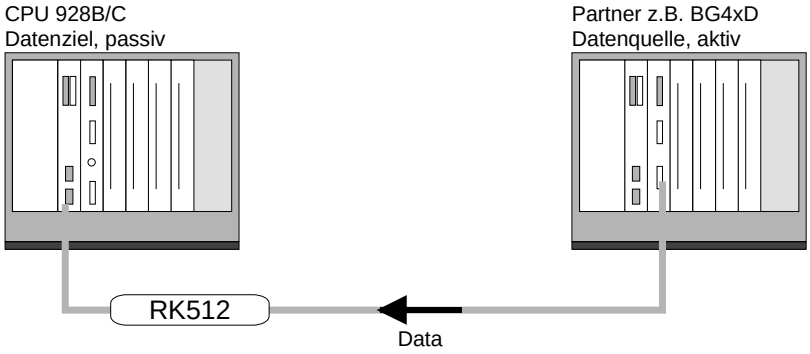
FETCH-Auftrag

Die CPU 928 ist aktiv, d.h. sie holt Daten vom Partner. Der Partner ist dabei passiv.



Partner-SEND-Auftrag

Der Partner ist aktiv, d.h. er sendet Daten zur CPU 928. Die CPU 928 ist passiv - sie empfängt Daten.

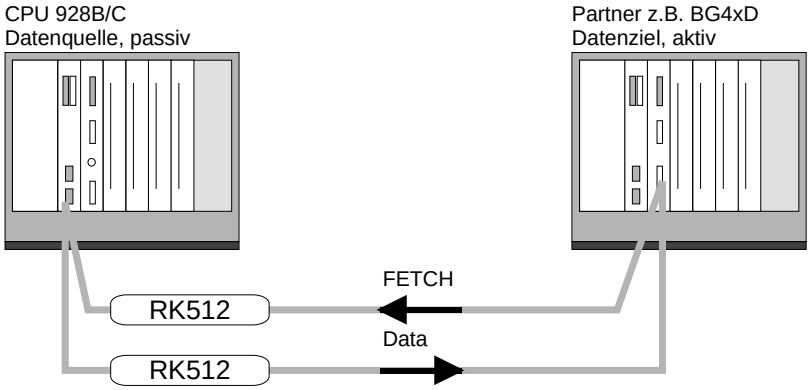


Hinweis!

Werden bei einem Partner-SEND-Auftrag Folgetelegramme empfangen, während sich die CPU im STOP-Zustand befindet, so wird erst das letzte Folgetelegramm mit Fehler 2Ah im Reaktionstelegramm quittiert.

Partner-FETCH-Auftrag

Der Partner ist aktiv, d.h. er holt Daten von der CPU 928. Die CPU ist dabei passiv.



Hinweis!

Sie können nur Datenwörter aus einem Datenbaustein (DB) oder einem erweiterten Datenbaustein (DX) übertragen.

Pro Auftrag sind maximal 2048 Datenwörter möglich. Sie können keine Ein-/ Ausgänge, Merker, Zeiten, Zähler etc. senden, holen oder empfangen.

Einbindung ins Anwenderprogramm

In diesem Abschnitt finden Sie eine Übersicht, wie die Rechnerkopplung RK 512 in ein Anwenderprogramm eingebunden wird.

Für das Erstellen der Anwenderprogramme wird Ihnen folgende Vorgehensweise empfohlen:

Vorabüberlegungen

- Mit welchem Partner soll die CPU Daten austauschen?
- Welches Schnittstellenmodul setzen Sie ein?
- Welche Daten sollen ausgetauscht werden?
- Wie sollen die Aufträge lauten?

Parametrierung der Schnittstelle

Zur Parametrierung der Schnittstelle benötigen Sie:

- DX 2
- statischen Parametersatz
- dynamische Parametersätze
- Auftragsfach
- Koordinierungsbyte Senden (KBS).

DX 2

In den erweiterten Datenbaustein DX 2 müssen Sie folgende Informationen eintragen:

- Kopplungstyp: Rechnerkopplung RK512
- Lage des statischen Parametersatzes
- Lage des ersten dynamischen Parametersatzes
- Lage des Auftragsfachs
- Lage des Koordinierungsbytes Senden (KBS)

Statischer Parametersatz

In den DB/DX mit dem statischen Parametersatz sind die Parameter zur Bit-Übertragungsschicht und zur Sicherungsschicht einzutragen.

Parameter der Bit-Übertragungsschicht (Schicht 1):

- Baudrate (Bit/S)
- Parität
- Bit pro Zeichen
- Stoppbits

Parameter der Sicherungsschicht (Schicht 2)

In der Sicherungsschicht werden die Eigenschaften der Übertragungsprozedur 3964(R) festgelegt:

- Modus: Prozedur 3964(R) mit Standardwerten oder parametrierbar
- Priorität
- Zeichenverzugszeit
- Quittungsverzugszeit
- Aufbauversuche
- Wiederholanzahl

Dynamische Parametersätze

In den DB/DX mit den dynamischen Parametersätzen müssen Sie die Auftrags-Parameter eintragen, die den Partner betreffen:

- SEND- oder FETCH-Auftrag
- Datenziel im Partner bei einem SEND-Auftrag oder Datenquelle im Partner bei einem FETCH-Auftrag
- Koordinierungsmerker (Byte- und Bit-Nr.) Nummer der CPU, wenn Sie mit mehreren CPUs Daten austauschen
- Nummer der CPU, wenn Sie über einen CP 524, CP 525 oder CP 544 mit mehren CPUs des Partners Daten austauschen

Auftragsfach

Tragen Sie hier die Auftrags-Parameter ein, die die eigene CPU betreffen:

- SEND- oder FETCH-Auftrag
- Auftragsnummer (1 bis 255) zur Auswahl des gewünschten dynamischen Parametersatzes
- Lage des Anzeigenwortes
- Datenquelle bei einem SEND-Auftrag oder Datenziel bei einem FETCH-Auftrag
- Länge der zu übertragenden Daten
- Lage des Parametrierfehlerbytes (PAFE)

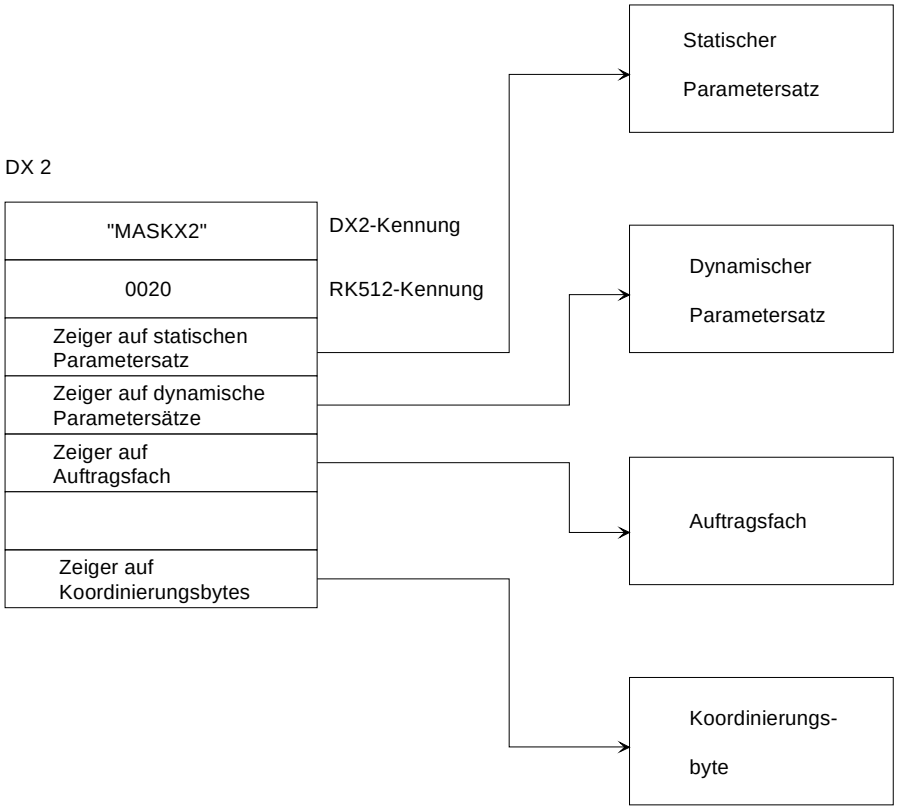
KBS Koordinierungsbyte Senden

Mit dem Koordinierungsbyte Senden (KBS) können Sie:

- einen SEND-Auftrag starten
- einen FETCH-Auftrag starten
- Die Rechnerkopplung RK 512 zurücksetzen.

Den Parametersatz im DX 2, den statischen Parametersatz und die dynamischen Parametersätze können Sie mit dem Programmiertool MC5 von VIPA erstellen.

Komponenten des Anwender-programms



CPU passiv

Bei einer ausschließlich passiv betriebenen CPU, die selbst keine Aufträge anstößt, benötigen Sie für die Rechnerkopplung RK 512 keine dynamischen Parametersätze und kein Auftragsfach.

Hinweis!

Ist in der CPU kein DX 2 vorhanden, wird die zweite Schnittstelle als PG-Schnittstelle initialisiert.

Die Parameter im DX 2, im statischen Parametersatz und in den dynamischen Parametersätzen, legen Sie fest, bevor Sie die Rechnerkopplung RK 512 in Betrieb nehmen.

Die Parameter im DX 2 werden nur bei Neustart vom Systemprogramm übernommen.

Die Parameter im statischen Parmetersatz (DW 0 bis DW 4) werden nur bei Spannungswiederkehr und bei Neustart vom Systemprogramm übernommen.

Die Parameter im statischen Parametersatz ab DW 5 werden nur bei Spannungswiederkehr, Neustart und dem Auftrag "Rechnerkopplung RK512 rücksetzen" vom Systemprogramm übernommen.

Die Parameter im Auftragsfach und im ausgewählten dynamischen Parametersatz werden zu Beginn eines Auftrages vom Systemprogramm übernommen. Diese Parameter sowie das Koordinierungsbyte Senden (KBS) dürfen Sie nur verändern, wenn kein Auftrag läuft.

Parametrierung

Für die Parametrierung können Sie das Programmiertool MC5 von VIPA einsetzen. Hiermit können Sie alle hier aufgeführten Bausteine parametrieren.

DX 2 parametrieren

Im DX 2 wird der Kopplungstyp vermerkt. Außerdem enthält er die Zeiger auf sämtliche benötigte Parameter und Parametersätze.

Die Parametrieung im DX 2 beginnt absolut bei DW 0. Alle Zahlenwerte sind in hexadezimaler Schreibweise dargestellt.

DW	Parameter	Bedeutung
0	4D41	MA
1	534B	SK
2	5832	X2
Kopplungstyp		
3	0020	Rechnerkopplung RK512
Zeiger auf statischen Parametersatz		
4	44xx oder 58xx	DB-Nr.xx oder [3] DX-Nr.xx
5	xxxx	ab DW-Nr. xxxx
Zeiger auf dynamische Parametersätze [1]		
6	44xx oder 58xx	DB-Nr. xx oder [3] DX-Nr. xx
7	xxxx	ab DW-Nr. xxxx
Zeiger auf Auftragsfach [1]		
8	44xx oder 58xx	DB-Nr. xx [3] DX-Nr.xx
9	xxxx	ab DW-Nr. xxxx
10 11 12 13	0000 0000 0000 0000	reserviert reserviert reserviert reserviert
Zeiger auf KBS [2]		
14	4D00 oder 5300 oder 44xx oder 58xx	Merker oder S-Merker oder DB-Nr. xx oder [3] DX-Nr. xx
15	xxxx	MW-NR., SW-Nr. oder DW-NR.

[1] Nicht notwendig, wenn die CPU nur passiv ist. Tragen Sie dann in die Datenwörter den Wert 0000 ein.

[2] Das KBS belegt im Speicher immer ein Wort.

[3] Wertebereich für xx von 3 bis FFh (3-255)

Beispiel:

In diesem Beispiel sehen Sie, wie Sie einen DX 2 mit folgenden Grunddaten parametrieren.

- Der statische Parametersatz liegt in DB 10 ab DW 4.
- Die dynamischen Parametersätze liegen im erweiterten DB 3 ab DW 4.
- Das Auftragsfach liegt in DB 6 ab DW 2.
- Das KBS liegt in MW 6.

Dies ergibt folgende Parametrierung:

Beispiel DX 2

```
0:      KH=4D41;    "MASKX2"
1:      KH=534B;    "MASKX2"
2:      KH=5832;    "MASKX2"
3:      KH=0020;    Kopplungstyp Rechnerkopplung RK512
4:      KH=440A;    Statischer Parametersatz in DB 10
5       KH=0004;    Ab DW 4
6       KH=5803;    Dynamische Parametersätze in DX 3
7       KH=0004;    Ab DW 4
8:      KH=4406;    Auftragsfach in DB 6
9       KH=0002;    Ab DW 2
10      KH=0000;    reserviert
11      KH=0000;    reserviert
12      KH=0000;    reserviert
13      KH=0000;    reserviert
14      KH=4D00;    KBS in Merker
15      KH=0006;    MW 6
16      KH=xxxx     gehört nicht mehr zur Parametrierung des
                    DX 2, beliebiger Inhalt, muss nicht
                    vorhanden sein.
```

Statischer Parametersatz

Im statischen Parametersatz müssen Sie die Parameter zur Bit-Übertragungsschicht und zur Sicherungsschicht eintragen.

In der nachfolgenden Tabelle steht, mit welchen Werten Sie den statischen Parametersatz in einem Datenbaustein oder erweiterten Datenbaustein hinterlegen können.

Die Parametrierung beginnt bei dem im DX 2 parametrierten DW.

Alle Zahlenwerte sind hexadezimal dargestellt.

DW (relativ)	Werte-Bereich	Bedeutung
Baudrate		
0	0001	110 Baud
	0002	150 Baud
	0003	300 Baud
	0004	600 Baud
	0005	1200 Baud
	0006	2400 Baud
	0007	4800 Baud
	0008	9600 Baud
	0009	19200 Baud
	000A	38400 Baud
Parität		
1	0000	keine (no)
	0001	ungerade (odd)
	0002	gerade (even)
Bit pro Zeichen		
2	0008	8 Bit pro Zeichen
Stoppbits		
3	0001	Stoppbit
	0002	1 + ½ Stoppbits
	0003	2 Stoppbits
-		
4	0000	reserviert
Modus (siehe Hinweis)		
5	0001	RK 512 mit Prozedur 3964 mit Standardwerten
	0002	RK 512 mit Prozedur 3964R mit Standardwerten
	0003	RK 512 mit Prozedur 3964 parametrierbar
	0004	RK 512 mit Prozedur 3964R parametrierbar
Priorität		
6	0000	niedrig
	0001	hoch

...Fortsetzung

...Fortsetzung

DW (relativ)	Werte-Bereich	Bedeutung
Zeichenverzugszeit (siehe Hinweis)		
7	0001 bis FFFF	Überwachungszeit 0,01 s bis 655,35 s (Einheit: 10ms)
Quittungsverzug (siehe Hinweis)		
8	0001 bis FFFF	Überwachungszeit 0,01 s bis 655,35 s (Einheit: 10ms)
Zeichenverzugszeit (siehe Hinweis)		
9	0001 bis 00FF	Anzahl
Wiederholanzahl (siehe Hinweis)		
10	0001 bis 00FF	Anzahl

Hinweis!

Die Datenwörter 7 bis 10 werden nur dann ausgewertet, wenn Sie als Modus 0003 oder 0004 eingetragen haben.

Vorsicht!

Wenn Sie als Modus 0001 oder 0002 eingetragen haben, werden die Standardwerte verwendet. Beachten Sie aber, dass die Datenwörter 7 bis 10 trotzdem vorhanden sein müssen.

	Modus 0001	Modus 0002
Zeichenverzugzeit	220 ms	220 ms
Quittungsverzugszeit	550ms	2000 ms
Aufbauversuche	6	6
Wiederholanzahl	6	6

Bedeutung der Parameter

Baudrate

Geschwindigkeit der Datenübertragung, Angabe in Bits/s (Baud).

Parität

Anzahl der Bits mit dem Wert "1" in einer einheitlichen langen Folge von Informationsbits (z.B. ein Byte). Die Parität ist, je nach Wert, "gerade" oder "ungerade". Zur Paritätskontrolle wird diese Folge von Informationsbits um ein weiteres Bit, das Paritätsbit (parity bit), erweitert, das durch seinen addierten Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt.

Bit pro Zeichen

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird (fest auf 8).

Stoppbits

Dauer der Stoppbits relativ zur Übertragungsdauer eines Informationsbits. Die Stoppbits werden bei der Start-Stopp-Übertragung jedem zu übertragenden Zeichen nachgesetzt.

Modus

Variante der möglichen Übertragungsprozeduren, 3964(R), jeweils mit Standardwerten oder parametrierbar.

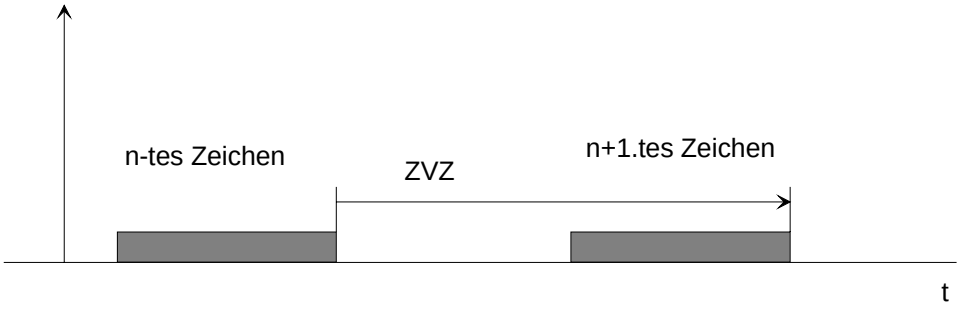
Priorität

Ein Partner hat Priorität, wenn sein Sendewunsch Vorrang vor dem Sendewunsch des anderen Partners hat.

Ein Partner hat niedere Priorität, wenn sein Sendewunsch hinter den Sendewunsch des anderen Partners zurückstehen muss. Beim Betrieb der Rechnerkopplung müssen Sie beide Koppelpartner mit verschiedener Priorität parametrieren, d.h. **A** bekommt Priorität =**1** (hoch), **B** bekommt Priorität = **0** (niedrig).

Zeichenverzugszeit (ZVZ)

Der maximal zulässige zeitliche Abstand zwischen zwei empfangenen Zeichen (siehe folgende Tabelle).



Bedeutung der Parameter

Quittungsverzugzeit (QVZ)

Die maximal zulässige Zeitspanne bis zur Quittung des Partners bei Verbindungsaufbau/-abbau.

Baudrate	kleinste zulässige ZVZ	kleinste zulässige QVZ
110 Baud	120 ms	120 ms
150 Baud	100 ms	100 ms
300 Baud	60 ms	60 ms
600 Baud	40 ms	40 ms
1200 Baud	30 ms	30 ms
2400 Baud	20 ms	20 ms
4800 Baud	20 ms	20 ms
9600 Baud	20 ms	20 ms
9600 Baud	20 ms	20 ms
19600 Baud	20 ms	20 ms
38400 Baud	20 ms	20 ms

Aufbauversuche

Maximale Anzahl der Versuche der CPU 928 eine Verbindung aufzubauen.

Wiederholanzahl

Maximale Anzahl der Telegrammwiederholungen (inklusive erstes Telegramm) bei Fehlern.

Beispiel

In diesem Beispiel sehen Sie, wie Sie einen statischen Parametersatz im DB 10 ab DW 4 parametrieren.

DB10

```
0:     KH =xxxx;     gehört nicht zum statischen
                     Paramertersatz, beliebiger Inhalt
1:     KH =xxxx;
2:     KH =xxxx;
3:     KH =xxxx;
4:     KH =0008;     Baudrate = 9600 Baud
5:     KH =0002;     Parität = 2 = Gerade
6:     KH =0008;     Bit pro Zeichen =8
7:     KH =0001;     Stoppbits = 1
8:     KH =0000;     reserviert
9:     KH =0003;     Modus = 3 = Prozedur 3964 parametrier-
                     bar
10:    KH =0001;     Priorität = 1 = hoch
11:    KH =00FF;     Zeichenverzugszeit = 255 x 0,01 s
                     (= 2,55 s)
12:    KH =010A;     Quittungsverzugzeit =266 x 0,01 s
                     (= 2,66 s)
13:    KH =0003;     Aufbauversuche = 3
14:    KH =0005;     Wiederholanzahl =5
15:    KH =xxxx;     gehört nicht mehr zum statischen Para-
                     metersatz, beliebiger Inhalt, muss
                     nicht vorhanden sein.
16:
  :
```

Bei der Rechnerkopplung RK 512 müssen die Parameter der Bit-Übertragungsschicht beim Partner mit denen der CPU 928 übereinstimmen.

Die Parameter der Sicherungsschicht müssen beim Partner wie folgt gesetzt werden:

- Verwendung der gleichen Prozedur wie bei der CPU 928
- entgegengesetzte Priorität wie bei der CPU 928

Dynamische Parametersätze

Die dynamischen Parametersätze enthalten die Auftrags-Parameter, die den **Partner** betreffen.

Ein dynamischer Parametersatz ist **3 Datenwörter** lang und liegt in einem DB. Ein DB kann maximal **255 dynamische Parametersätze** enthalten. Jedem dynamischen Parametersatz ist eine Auftragsnummer zugeordnet. Der erste dynamische Parametersatz (rel. DW 0 bis DW 2) hat die Auftragsnummer 1, der zweite (rel. DW 3 bis DW 5) die Auftragsnummer 2 usw.

Im DXs 2 geben Sie die Lage des ersten dynamischen Parametersatzes (A.-NR. 1) an. Bevor Sie einen SEND- oder FETCH-Auftrag anstoßen, müssen Sie den dazugehörigen dynamischen Parametersatz parametriert haben. Mit der Auftragsnummer, die Sie im Auftragsfach angeben, wird zu jedem SEND- oder FETCH-Auftrag der gewünschte dynamische Parametersatz angewählt.

Belegung

In der nachfolgenden Tabelle steht, mit welchen Werten Sie die dynamischen Parametersätze in einem DB oder erweiterten DB hinterlegen können. Die Parametrierung beginnt bei dem im DX 2 parametrierten DW.

Alle Zahlenwerte sind in hexadezimaler Schreibweise dargestellt.

A-NR.	DW (rel.)	Parameter	Bedeutung
SEND-/FETCH-Auftrag mit der Auftragsnummer 1			
1	Auftragstyp		
	0	4144 oder 4158 oder 4544 oder 4558	SEND-Auftrag mit Datenziel DB oder SEND-Auftrag mit Datenziel DX oder FETCH-Auftrag mit Datenquelle DB oder FETCH-Auftrag mit Datenquelle DX
	Datenziel (SEND) / Datenquelle (FETCH)		
	1	xxyy	DB/DX-NR. xx ab DW-Nr.yy (zulässige Bereiche s.u.)
	Koordinierungsmerker (KM)		
	2	xxyz	Byte-Nr. **xx** des Koordinierungsmerkers [*1] Wertebereich 00 bis FF (0 bis 255) (siehe Hinweis) Bit-NR. **z** des Koordinierungsmerkers Wertebereich 0 bis 7 oder F CPU-Nummer **y** Wertebereich 1 bis 4 (bzw. 0,F)
SEND-/FETCH-Auftrag mit der Auftragsnummer 2			
2	3 4 5		
. . .	. . .	. . .	. . .
SEND-/FETCH-Auftrag mit der Auftragsnummer 255			
255	762 763 764		

[*1] Wird kein Koordinierungsmerker angegeben, so müssen Sie als Byte-Nr. FF und als Bit-Nr. F eintragen.

In der nächsten Tabelle stehen die bei einem SEND-Auftrag zulässigen Datenbausteine und Datenwörter für das Datenziel.

SEND-Auftrag		Grenzwert für die Parametrierung eines dynamischen Parametersatzes			
Datenquelle (CPU 928B)	Datenziel (Koppel-partner)	Auftragstyp	Datenziel		KM erlaubt
			DB-Nr. DX-Nr.	ab DW-Nr.	
DB	DB	AD	3-255	0-255	ja
DX	DB	AD	3-255	0-255	ja
DB	DX	AX	3-255	0-255	ja
DX	DX	AX	3-255	0-255	ja

In der nächsten Tabelle stehen die bei einem FETCH-Auftrag zulässigen Datenbausteine und Datenwörter für die Datenquelle.

FETCH-Auftrag		Grenzwert für die Parametrierung eines dynamischen Parametersatzes			
Datenquelle (CPU 928B)	Datenziel (Koppel-partner)	Auftragstyp	Datenziel		KM erlaubt
			DB-Nr. DX-Nr.	ab DW-Nr.	
DB	DB	AD	3-255	0-255	ja
DX	DB	AD	3-255	0-255	ja
DB	DX	AX	3-255	0-255	ja
DX	DX	AX	3-255	0-255	ja

Beispiel

In folgendem Beispiel sehen Sie, wie Sie die dynamischen Parametersätze mit den Auftragsnummern 2 und 3 parametrieren können. Die dynamischen Parametersätze liegen im erweiterten Datenbaustein DX 3 ab DW 4.

```
0:     KH =xxxx;     gehört nicht zu den dynamischen
                     Paramertersätzen, beliebiger Inhalt
1:     KH =xxxx;
2:     KH =xxxx;
3:     KH =xxxx;
4:     KH =yyyy;     Auftrag mit der Auftragsnummer 1
5:     KH =yyyy;     beliebiger Inhalt, wenn die Auftrags-
                     nummer 1 nicht
6:     KH =yyyy;     benötigt wird
7:     KH =4144;     Auftrag mit der Auftragsnummer 2
8:     KH =0B05;     SEND-Auftrag mit Datenziel DB11 ab DW5
9:     KH =6412;     in CPU 1 mit KM 100.2
10:    KH =4558;     Auftrag mit der Auftragsnummer 3
11:    KH =110D;     FETCH-Auftrag mit Datenquelle DX 17 ab
                     DW 13
12:    KH =FFFF;     von einer CPU des Koppelpartners ohne
                     KM
13:    KH =yyyy;     Auftrag mit der Auftragsnummer 4
14:    KH =yyyy;     beliebiger Inhalt, muss nicht
                     vorhanden sein, wenn keine Aufträge
                     mit einer Auftragsnummer ≥4 benötigt
                     werden.
15:
```

Auftragsfach

In der nachfolgenden Tabelle steht mit welchen Werten Sie das Auftragsfach in einem DB oder erweiterten DB hinterlegen können. Die Parameter im Auftragsfach dürfen Sie nur dann verändern, wenn kein Auftrag läuft.

Die Parametrierung beginnt bei dem im DX 2 parametrierten DW.

Alle Zahlenwerte sind in hexadezimaler Schreibweise dargestellt.

DW (rel.)	Parameter	Bedeutung
SEND-FETCH-Auftrag		
0	41xx oder	A-SEND-Auftrag mit der Auftragsnummer xx (1 bis 255) oder
	45xx oder	E-FETCH-Auftrag mit Auftragsnummer xx (1 bis 255) oder
Anzeigenwort (ANZW) *1		
1	4D00 oder	im Merkerbereich oder
	5300 oder	im S-Merkerbereich oder
	44xx oder	im Datenbaustein DB-Nr. xx oder
	58xx	im erweiterten DB DX-Nr. xx
2	xxxx	MW-Nr., SW-Nr. oder DW-Nr.
Datenquelle / Datenziel *2		
3	44xx oder	im Datenbaustein DB-Nr. xx oder
	58xx	im erweiterten Datenbaustein DX-Nr. xx
4	xxxx	ab DW-Nr. xxxx
5	xxxx	Länge der zu übertragenden Daten in Wörtern
Datenquelle / Datenziel *2		
3	44xx oder	im Datenbaustein DB-Nr. xx oder
		im erweiterten Datenbaustein DX-Nr. xx
4	xxxx	ab DW-Nr.xxxx
5	xxxx	Länge der zu übertragenden Daten in Wörtern
Parametrierfehlerbyte (PAFE - Byte)*1		
6	4D00 oder	im Merkerbereich oder
	5300 oder	im S-Merkerbereich oder
	44xx oder	im Datenbaustein DB-Nr. xx oder
	58xx	im erweiterten Datenbaustein DX-Nr. xx
7	xxxx	

*1 Detailierte Informationen zum Anzeigenwort (ANZW) und zum Parametrierfehlerbyte (PAFE-Byte) stehen im folgenden Abschnitt.

*2 Die zulässigen Werte stehen in den nachfolgenden Tabellen.

In der nächsten Tabelle stehen die bei einem SEND-Auftrag zulässigen Datenbausteine, Datenwörter und Längenangaben für die Datenquelle.

SEND-Auftrag		Grenzwert für die Parametrierung des Auftragsfachs			
Datenquelle (CPU 928)	Datenziel (Koppel-partner)	Datenquelle	Datenquelle		Länge (Wörter)
			DB-Nr. DX-Nr.	ab DW-Nr.	
DB	DB	DB	3-255	0-2047	1-2048
DX	DB	DX	3-255	0-2047	1-2048
DB	DX	AX	3-255	0-2047	1-2048
DX	DX	AX	3-255	0-2047	1-2048

Die bei einem FETCH-Auftrag zulässigen Datenbausteine, Datenwörter und Längenangaben für das Datenziel.

FETCH-Auftrag		Grenzwert für die Parametrierung des Auftragsfachs			
Datenziel (CPU 928B)	Datenquelle (Koppel-partner)	Datenziel	Datenziel		Länge (Wörter)
			DB-Nr. DX-Nr.	ab DW-Nr.	
DB	DB	DB	3-255	0-2047	1-2048
DB	DX	DB	3-255	0-2047	1-2048
DX	DB	DX	3-255	0-2047	1-2048
DX	DX	DX	3-255	0-2047	1-2048

Beispiel

wie Sie ein Auftragsfach im DB 6 ab DW 2 parametrieren können.

DB 6

```
0:      KH =xxxx;     gehört nicht ins Auftragsfach ,
                      beliebiger Inhalt
1:      KH =xxxx;
2:      KH =4503;     FETCH-Auftrag, Auftragsnummer 3
3:      KH =4D00;     Anzeigenwort in Merker
4:      KH =0008;     Merkerwort 8
5:      KH =5813;     Datenziel in DX 19
6:      KH =0014;     ab DW 20
7:      KH =0032;     Länge = 50 Wörter
8:      KH =5300;     PAFE-Byte in S-Merker
9:      KH =0013;     S-Merkerbyte 19
10:     KH =xxxx;     gehört nicht mehr zum Auftratsfach,
11:     KH =xxxx;     beliebiger Inhalt muss nicht vorhanden
                      sein.
12:
```

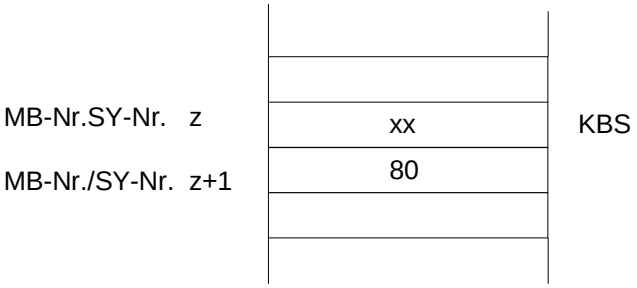
Koordinierungs-byte Senden (KBS)

Mit dem KBS können Sie folgende Aufträge starten:

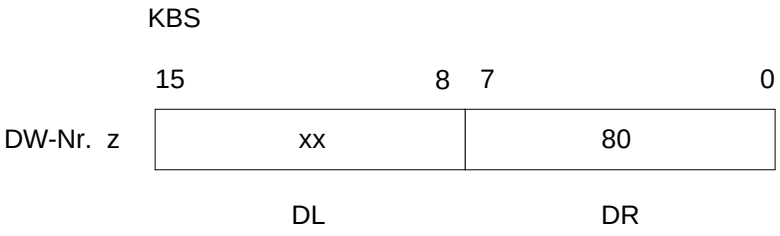
- **SEND**-Auftrag
- **FETCH**-Auftrag
- die **Rechnerkopplung RK 512** zurücksetzen

Das KSB kann Merker bzw. S-Merker sein oder im einem Datenbaustein bzw. erweiterten Datenbaustein liegen. Merker und S-Merker sind in der CPU 928B vorhanden, Datenbausteine und erweiterte Datenbausteine müssen Sie anlegen.

Darstellung im Merker- bzw. S-Merkerbereich:



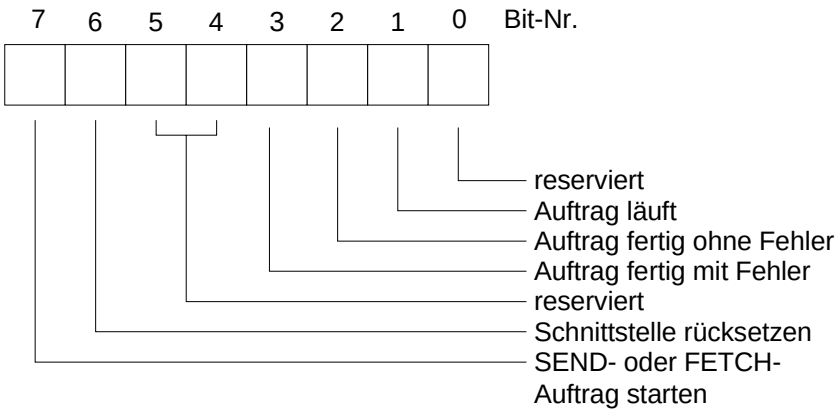
Darstellung im DB bzw. im erweiterten DB:



Hinweis!

Für das Koordinierungsbyte Senden (KBS) wird vom Systemprogramm ein DW bzw. MW reserviert. Das nicht benötigte Byte wird bei Spannungs-wiederkehr und Neustart mit 80h belegt.

Das Koordinierungsbyte Senden (KBS) ist folgendermaßen aufgebaut:



Die Bits haben folgende Bedeutung:

Bit 1 "Auftrag läuft"

- wird vom Systemprogramm gesetzt, wenn ein Auftrag gestartet ist
- wird vom Systemprogramm zurückgesetzt, wenn der Auftrag beendet ist.
- können Sie auswerten, wenn Sie wissen wollen, ob es sinnvoll ist einen neuen Auftrag zu starten.

Bit 2 "Auftrag fertig ohne Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag ohne Fehler abgeschlossen wurde.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag das Bit 1 "Auftrag läuft" gesetzt wird.
- können Sie auswerten, wenn Sie wissen wollen, ob der Auftrag ohne Fehler abgeschlossen wurde.

Bit 3 "Auftrag fertig mit Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag mit Fehler abgeschlossen wurde.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag das Bit 1 "Auftrag läuft" gesetzt wird.
- können Sie auswerten, wenn Sie wissen wollen, ob der Auftrag mit Fehler abgeschlossen wurde.

Bit 6 "Schnittstelle zurücksetzen"

- können Sie setzen, wenn Sie die Schnittstelle zurücksetzen wollen.
- wird vom Systemprogramm zurückgesetzt, sobald das Systemprogramm die Schnittstelle rückgesetzt hat.

Bit 7 "SEND- oder FETCH-Auftrag starten"

- können Sie zum Starten eines SEND- oder FETCH-Auftrags setzen
- wird vom Systemprogramm zurückgesetzt, sobald das Systemprogramm den SEND- oder FETCH-Auftrag beendet hat.

Wenn Sie die Schnittstelle zurücksetzen, werden die Parameter der Sicherungsschicht des statischen Parametersatzes (ab DW 5) neu übernommen:

- Modus
- Priorität
- Zeichenverzugszeit
- Quittungsverzugszeit
- Aufbauversuche
- Wiederholversuche

Haben Sie KBS-Bit 6 und KBS-Bit 7 gleichzeitig gesetzt oder im zeitlichen Abstand von ≤ 10ms bei der CPU 928, zuerst KBS-Bit 6 und dann KBS-Bit 7 gesetzt, so führt das Systemprogramm zuerst das Rücksetzen, danach den SEND-FETCH-Auftrag aus.

Haben Sie im zeitlichen Abstand von ≤ 10ms bei der CPU 928 zuerst KBS-Bit 7 und dann KBS-Bit 6 gesetzt, wird die Reihenfolge der beiden Aufträge nicht garantiert.

Mit dem Rücksetzen brechen Sie einen laufenden Partner-SEND-/FETCH-Auftrag ab.

Hinweis!

Die Parameter der Bit-Übertragungsschicht der Übertragungsleitung (statischer Parametersatz bis einschließlich DW 4) können Sie nur durch einen Neustart der CPU verändern.

Ein Rücksetzen der Schnittstelle reicht dazu nicht aus.

Partner-Auftrage können Sie nicht vom Anwenderprogramm steuern.

Das Systemprogramm arbeitet sie selbständig im Hintergrund ab.

Das Systemprogramm legt die Daten (beim Partner-SEND) im Ziel-DB/DX ab, bzw. holt die Daten (beim Partner-FETCH)aus dem Quell-DB/DX. Nur mit Hilfe der Koordinierungsmerker können Sie den Datenaustausch bei Partner-Aufträgen kontrollieren.

Koordinierungsmerker (KM)

Mit Hilfe der Koordinierungsmerker können Sie das Überschreiben von Daten bei einem Partner-SEND-Auftrag sperren oder freigeben. Ebenso können Sie eine Übergabe von Daten bei einem Partner-FETCH-Auftrag sperren oder freigeben. Damit ist eine konforme Datenübertragung möglich.

Wertebereich des Koordinierungsmerkers:

Byte-Nr.:	0 bis 255
Bit-Nr.:	0 bis 7

Hinweis!

Überprüfen Sie bei der Parametrierung eines Koordinierungsmerkers, ob der Koppelpartner diesen auch auswerten kann.

Haben Sie z.B. einen CP544 oder CP 524/525 als Koppelpartner, so darf die Bytenummer des Koordinierungsmerkers nur die Werte 1 bis 223 annehmen.

Jeder Koordinierungsmerker belegt ein Merkerbit im Merkerbereich der CPU. Sie können keine S-Merker als Koordinierungsmerker benutzen.

Die Daten werden zwischen den Partnern in Form von Telegrammen ausgetauscht. Sendet der Partner ein SEND- oder FETCH-Telegramm mit einem Koordinierungsmerker an die CPU, überprüft die CPU, ob der Koordinierungsmerker bei ihr gesetzt ist.

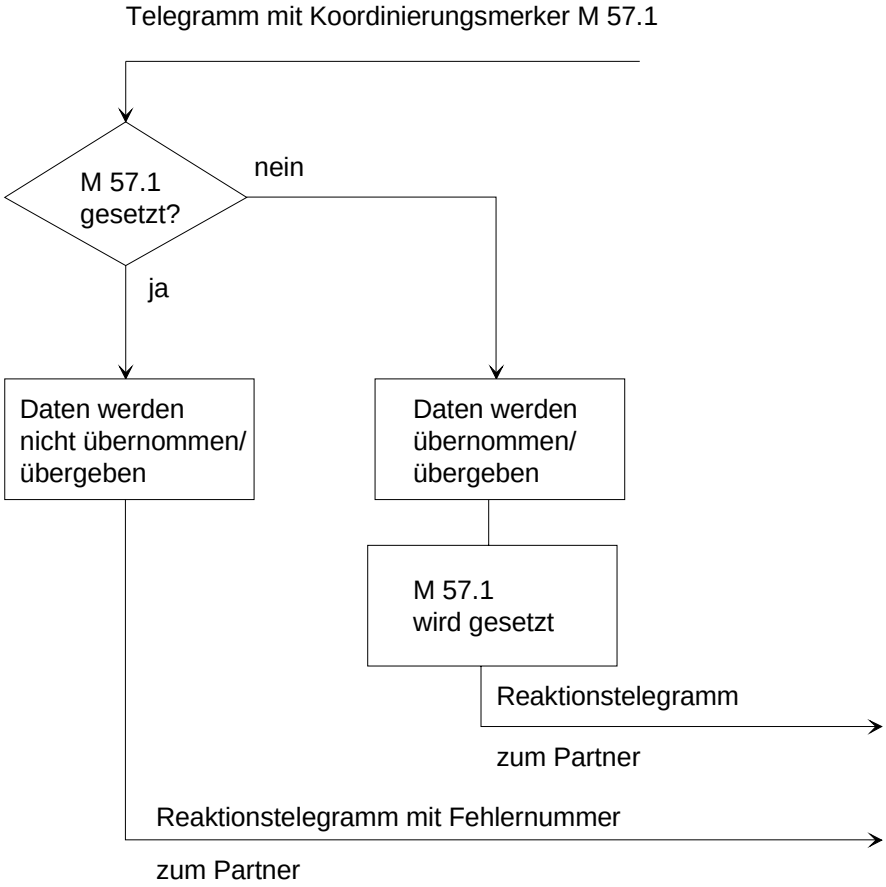
Ist der Koordinierungsmerker **nicht gesetzt**, werden die Daten des Partners übernommen oder an den Partner übergeben. Danach setzt das Systemprogramm den Koordinierungsmerker.

Ist der Koordinierungsmerker gesetzt, lehnt die CPU die Datenübergabe mit dem Partner ab und sendet ein Reaktionstelegramm mit Fehlernummer an den Partner zurück.

Hinweis!

Sie müssen nach dem Auswerten der Daten den Datenbereich wieder freigeben, indem Sie den Koordinierungsmerker im Anwenderprogramm wieder zurücksetzen.

Beispiel



Achtung!

Sonderfall FETCH:

Nach dem Setzen des Koordinierungsmerkers durch das Systemprogramm müssen die Daten vom Kommunikationsprozessor übertragen werden.

Der gesetzte Koordinierungsmerker garantiert nicht, dass der Koppelpartner die Daten auch fehlerfrei empfangen hat.

Inbetriebnahme

Voraussetzung

Sie haben das Schnittstellenmodul in die CPU eingebaut, die CPU steckt im ZG-Rahmen und die Spannung ist ausgeschaltet.

Vorgehensweise

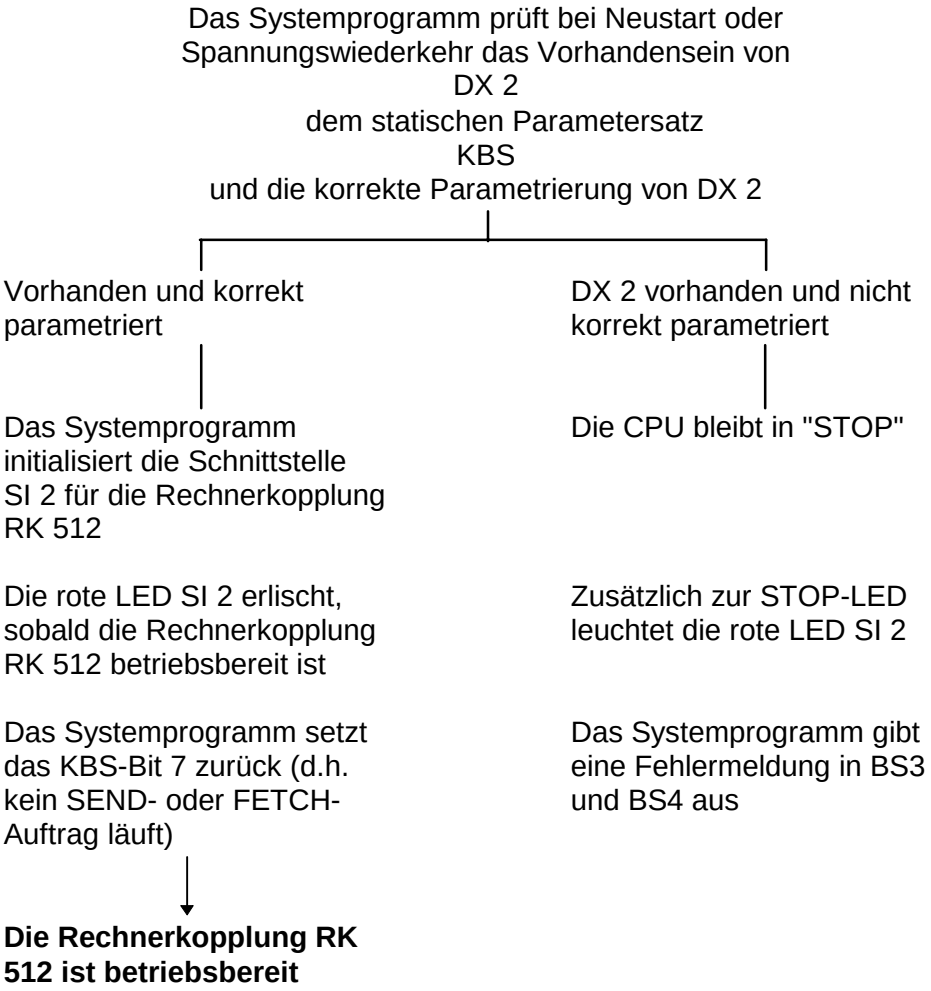
Für die Inbetriebnahme der Rechnerkopplung RK 512 empfehlen wir folgende Vorgehensweise:

Schritt	Handlung
1	Verbinden Sie die CPU mit dem Partner (mit einer Standard-Steckleitung).
2	Schalten Sie die Versorgungsspannung ein.
3	Übertragen Sie Ihr Anwenderprogramm zur CPU. Das Anwenderprogramm muss folgende **Datenbausteine** oder **erweiterte Datenbausteine** enthalten, die die Schnittstelle SI 2 als RK 512 parametrieren: DX 2 DB/DX mit dem statischen Parametersatz DB/DX mit dem dynamischen Parametersätzen *1 DB/DX mit dem Auftragsfach *1 DB/DX mit dem KBS (nur wenn das KBS in einem DB/DX liegt) *2
4	Führen Sie einen **Neustart** durch.
5	Bereiten Sie den Partner ebenfalls auf die Rechnerkopplung RK 512 vor.

*1 Nur wenn die CPU selbst aktive Aufträge anstößt

*2 Falls KBS ein MW oder S-MW ist, ist es bereits in der CPU vorhanden.

Ablauf


Das Systemprogramm prüft bei Neustart oder Spannungswiederkehr das Vorhandensein von
DX 2
dem statischen Parametersatz
KBS
und die korrekte Parametrierung von DX 2
Vorhanden und korrekt parametriert
DX 2 vorhanden und nicht korrekt parametriert
Das Systemprogramm initialisiert die Schnittstelle SI 2 für die Rechnerkopplung RK 512
Die CPU bleibt in "STOP"
Die rote LED SI 2 erlischt, sobald die Rechnerkopplung RK 512 betriebsbereit ist
Zusätzlich zur STOP-LED leuchtet die rote LED SI 2
Das Systemprogramm setzt das KBS-Bit 7 zurück (d.h. kein SEND- oder FETCH-Auftrag läuft)
Das Systemprogramm gibt eine Fehlermeldung in BS3 und BS4 aus
Die Rechnerkopplung RK 512 ist betriebsbereit

Betrieb

Voraussetzung

Die Schnittstelle SI 2 ist für die Rechnerkopplung RK 512 parametriert.

Die LED SI 2 leuchtet nicht. Die RK 512 ist betriebsbereit.

Arbeitsweise

Das Systemprogramm prüft jetzt ständig, ob das Koordinierungsbyte Senden (KBS) vorhanden ist. Ist es nicht vorhanden (z.B. weil der DB/DX gelöscht wurde), ruft das Systemprogramm den OB 35 auf.

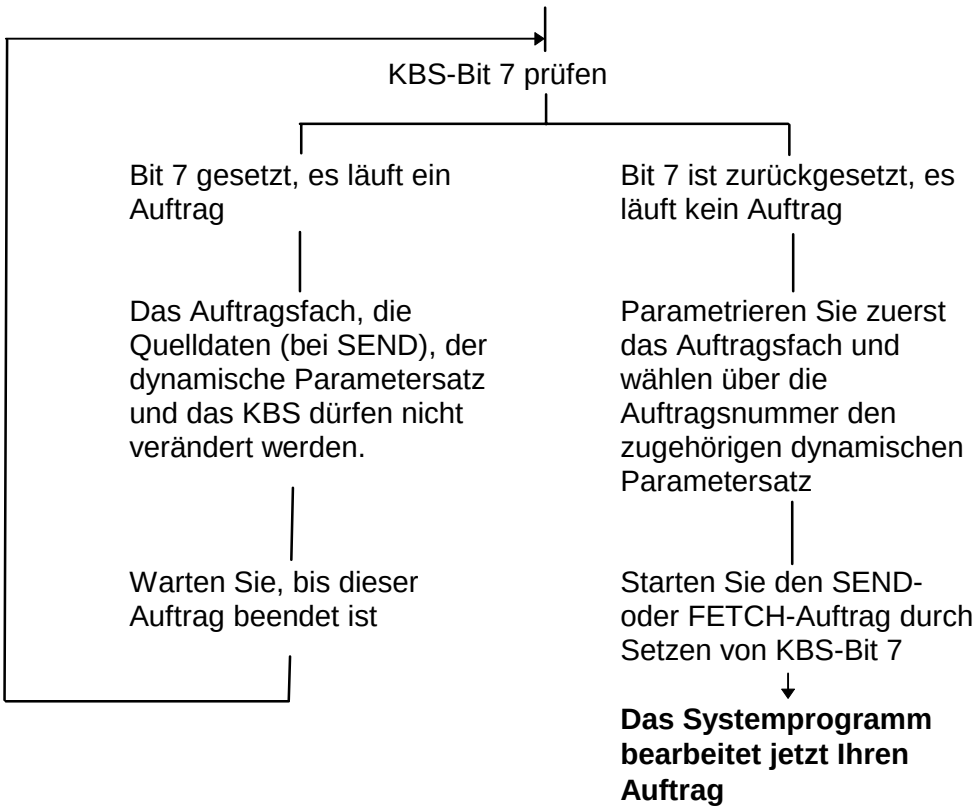
Die Fehlernummer wird in AKKU 1 eingetragen.

Das Systemprogramm bearbeitet jetzt die Aufträge, die Sie in Ihrem Anwenderprogramm programmiert haben, bzw. die Aufträge, die von Ihrem Partner angestoßen werden. Wie Sie die Aufträge in Ihrem Anwenderprogramm programmieren müssen, erfahren Sie in diesem Kapitel.

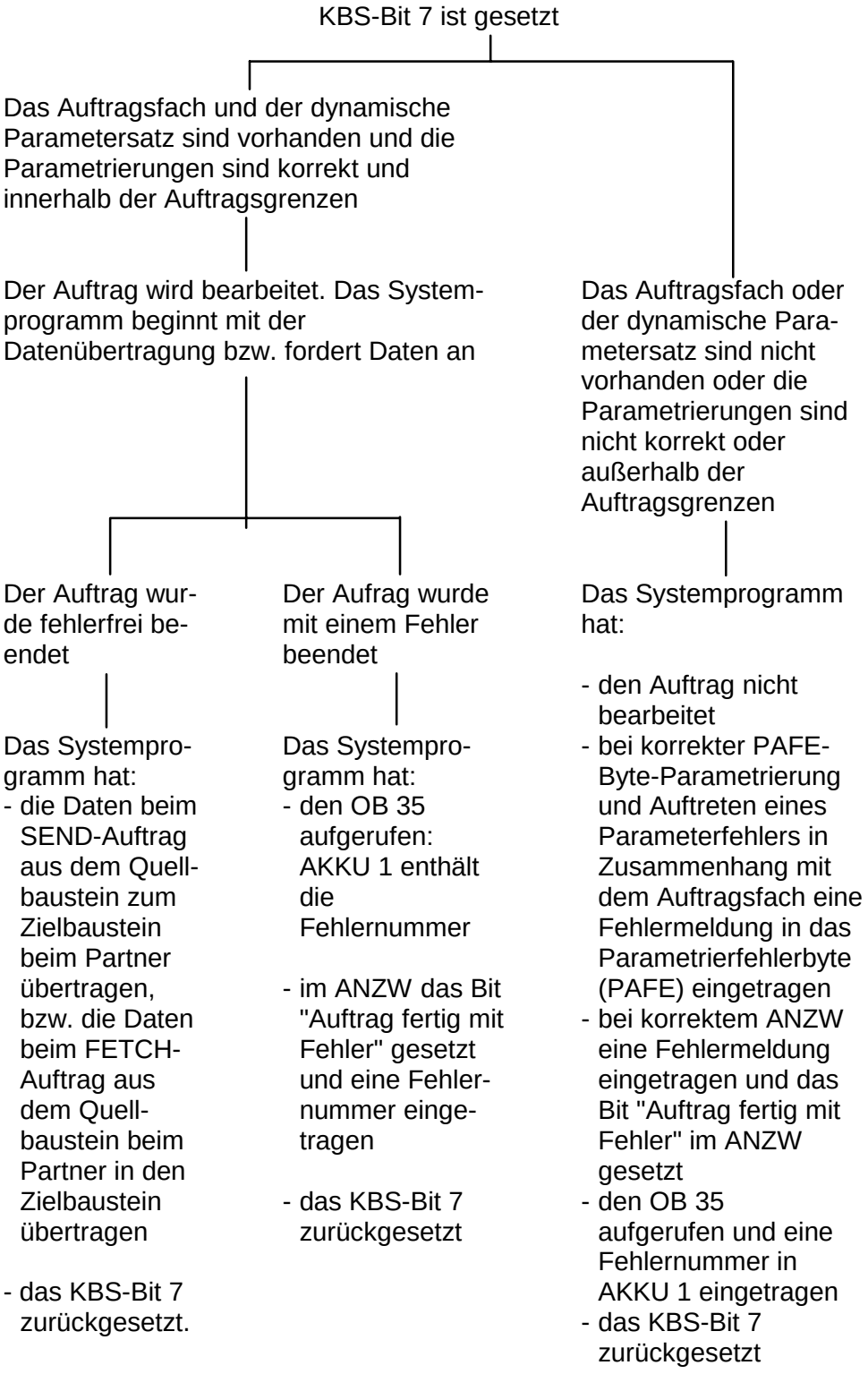
SEND/FETCH-Auftrag

Wenn Sie einen SEND/FETCH-Auftrag programmieren wollen, beachten Sie folgenden Ablauf.

Auftrag starten


KBS-Bit 7 prüfen
Bit 7 gesetzt, es läuft ein Auftrag
Bit 7 ist zurückgesetzt, es läuft kein Auftrag
Das Auftragsfach, die Quelldaten (bei SEND), der dynamische Parametersatz und das KBS dürfen nicht verändert werden.
Parametrieren Sie zuerst das Auftragsfach und wählen über die Auftragsnummer den zugehörigen dynamischen Parametersatz
Warten Sie, bis dieser Auftrag beendet ist
Starten Sie den SEND- oder FETCH-Auftrag durch Setzen von KBS-Bit 7
Das Systemprogramm bearbeitet jetzt Ihren Auftrag

Das nachfolgende Diagramm zeigt, welche Reaktionen auftreten können, wenn das KSB-Bit 7 gesetzt ist und das Systemprogramm Ihren Auftrag bearbeitet.

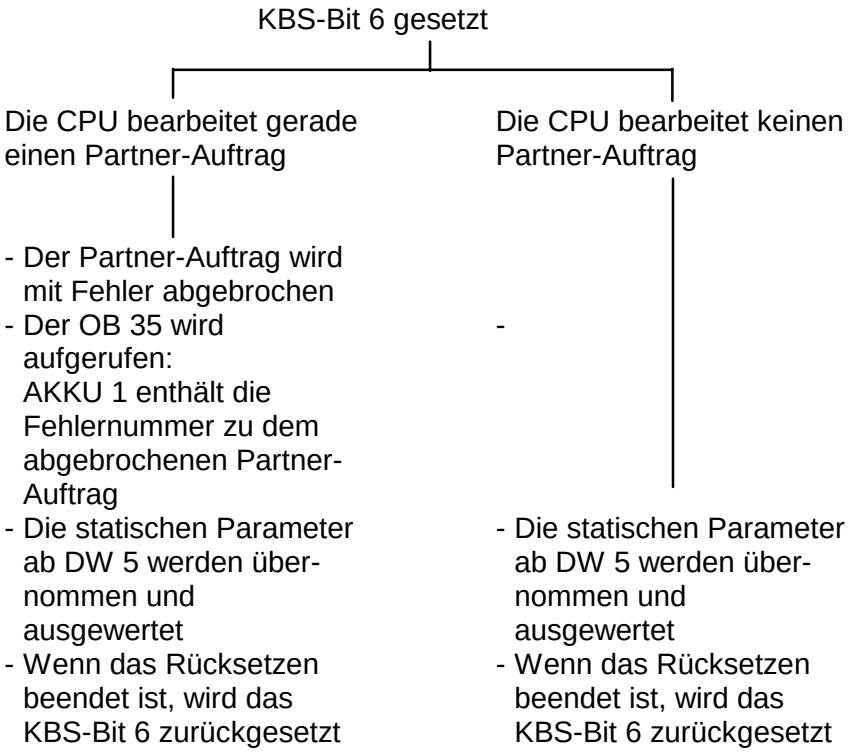

KBS-Bit 7 ist gesetzt
Das Auftragsfach und der dynamische Parametersatz sind vorhanden und die Parametrierungen sind korrekt und innerhalb der Auftragsgrenzen
Der Auftrag wird bearbeitet. Das Systemprogramm beginnt mit der Datenübertragung bzw. fordert Daten an
Das Auftragsfach oder der dynamische Parametersatz sind nicht vorhanden oder die Parametrierungen sind nicht korrekt oder außerhalb der Auftragsgrenzen
Der Auftrag wurde fehlerfrei beendet
Der Aufrag wurde mit einem Fehler beendet
Das Systemprogramm hat:
- die Daten beim SEND-Auftrag aus dem Quellbaustein zum Zielbaustein beim Partner übertragen, bzw. die Daten beim FETCH-Auftrag aus dem Quellbaustein beim Partner in den Zielbaustein übertragen
- das KBS-Bit 7 zurückgesetzt.
Das Systemprogramm hat:
- den OB 35 aufgerufen: AKKU 1 enthält die Fehlernummer
- im ANZW das Bit "Auftrag fertig mit Fehler" gesetzt und eine Fehlernummer eingetragen
- das KBS-Bit 7 zurückgesetzt
Das Systemprogramm hat:
- den Auftrag nicht bearbeitet
- bei korrekter PAFE-Byte-Parametrierung und Auftreten eines Parameterfehlers in Zusammenhang mit dem Auftragsfach eine Fehlermeldung in das Parametrierfehlerbyte (PAFE) eingetragen
- bei korrektem ANZW eine Fehlermeldung eingetragen und das Bit "Auftrag fertig mit Fehler" im ANZW gesetzt
- den OB 35 aufgerufen und eine Fehlernummer in AKKU 1 eingetragen
- das KBS-Bit 7 zurückgesetzt

Hinweis!

Sie dürfen erst dann einen neuen Auftrag starten, wenn der vorangegangene beendet ist.

Schnittstelle rücksetzen

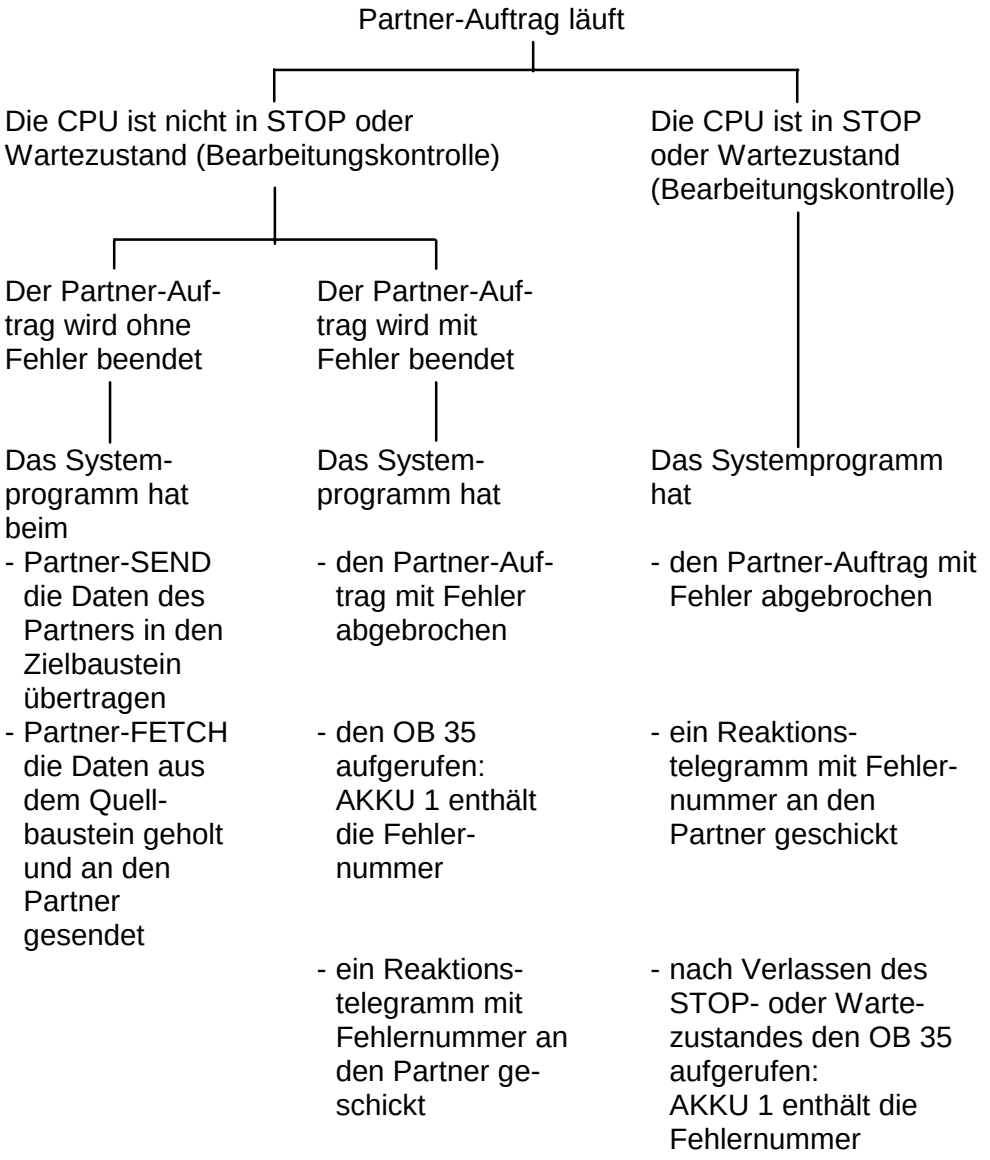
Sie haben das KBS-Bit 6 gesetzt und damit einen Rücksetz-Auftrag abgegeben. Das nachfolgende Diagramm zeigt die möglichen Reaktionen.


KBS-Bit 6 gesetzt
Die CPU bearbeitet gerade einen Partner-Auftrag
Die CPU bearbeitet keinen Partner-Auftrag
- Der Partner-Auftrag wird mit Fehler abgebrochen
- Der OB 35 wird aufgerufen: AKKU 1 enthält die Fehlernummer zu dem abgebrochenen Partner-Auftrag
- Die statischen Parameter ab DW 5 werden übernommen und ausgewertet
- Wenn das Rücksetzen beendet ist, wird das KBS-Bit 6 zurückgesetzt
-
- Die statischen Parameter ab DW 5 werden übernommen und ausgewertet
- Wenn das Rücksetzen beendet ist, wird das KBS-Bit 6 zurückgesetzt

Partner-Auftrag

Partner-Aufträge werden nicht vom Anwenderprogramm sondern vom Systemprogramm selbständig im Hintergrund bearbeitet. Um sicherzugehen, dass das Systemprogramm dabei nur mit konformen Daten arbeitet, können Sie in Ihrem Anwenderprogramm mit Hilfe der Koordinierungsmerker verhindern, dass Daten überschrieben (Partner-SEND) oder gelesen (Partner-FETCH) werden.

Das folgende Diagramm zeigt, welche Situationen nach Ablauf eines Partner-Auftrages auftreten kann


Partner-Auftrag läuft
Die CPU ist nicht in STOP oder Wartezustand (Bearbeitungskontrolle)
Die CPU ist in STOP oder Wartezustand (Bearbeitungskontrolle)
Der Partner-Auftrag wird ohne Fehler beendet
Der Partner-Auftrag wird mit Fehler beendet
Das Systemprogramm hat beim
- Partner-SEND die Daten des Partners in den Zielbaustein übertragen
- Partner-FETCH die Daten aus dem Quellbaustein geholt und an den Partner gesendet
Das Systemprogramm hat
- den Partner-Auftrag mit Fehler abgebrochen
- den OB 35 aufgerufen: AKKU 1 enthält die Fehlernummer
- ein Reaktionstelegramm mit Fehlernummer an den Partner geschickt
Das Systemprogramm hat
- den Partner-Auftrag mit Fehler abgebrochen
- ein Reaktionstelegramm mit Fehlernummer an den Partner geschickt
- nach Verlassen des STOP- oder Wartezustandes den OB 35 aufgerufen: AKKU 1 enthält die Fehlernummer

Verhalten bei speziellen Betriebs-zuständen

Stoppzustand

Im Betriebszustand "STOP" findet keine Anwenderprogrammbearbeitung statt, d.h. Sie können keinen neuen Auftrag starten. Beim Übergang vom Betriebszustand "RUN" in den Betriebszustand "STOP" gilt für einen bereits laufenden, vom Anwenderprogramm angestoßenen Auftrag:

- Der Auftrag wird mit Fehler abgebrochen, wenn noch Daten zwischen dem Betriebssystem-Prozessor und dem Dual-Port-RAM transferiert werden müssen.
- Der Auftrag wird normal beendet, wenn keine Daten mehr zwischen dem Betriebssystem-Prozessor und dem Dual-Port-RAM transferiert werden müssen.

Stößt der Koppelpartner im Betriebszustand "STOP" einen Auftrag an, wird dieser negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Beim Übergang vom Betriebszustand "RUN" in den Betriebszustand "STOP" wird ein bereits laufender, vom Koppelpartner angestoßener Auftrag mit Fehler abgebrochen, d. h. negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Warte-zustand

Die CPU nimmt bei der PG-Online-Funktion "BEARBEITUNGS-KONTROLLE" den Betriebszustand "Wartezustand" ein. Im "Wartezustand" findet keine Anwenderprogrammbearbeitung statt, d.h. Sie können keinen neuen Auftrag starten. Ein bereits laufender, vom Anwenderprogramm angestoßener Auftrag wird nach Beendigung des Betriebszustandes "Wartezustand" fortgesetzt und abgeschlossen.

Stößt der Koppelpartner im Betriebszustand "Wartezustand" einen Auftrag an, wird dieser negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen. Bei Eintritt in den Betriebszustand "Wartezustand" wird ein bereits laufender, vom Koppelpartner angestoßener Auftrag mit Fehler quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Fehlerbe-arbeitung

Fehler werden Ihnen durch eine Fehlernummer gemeldet:

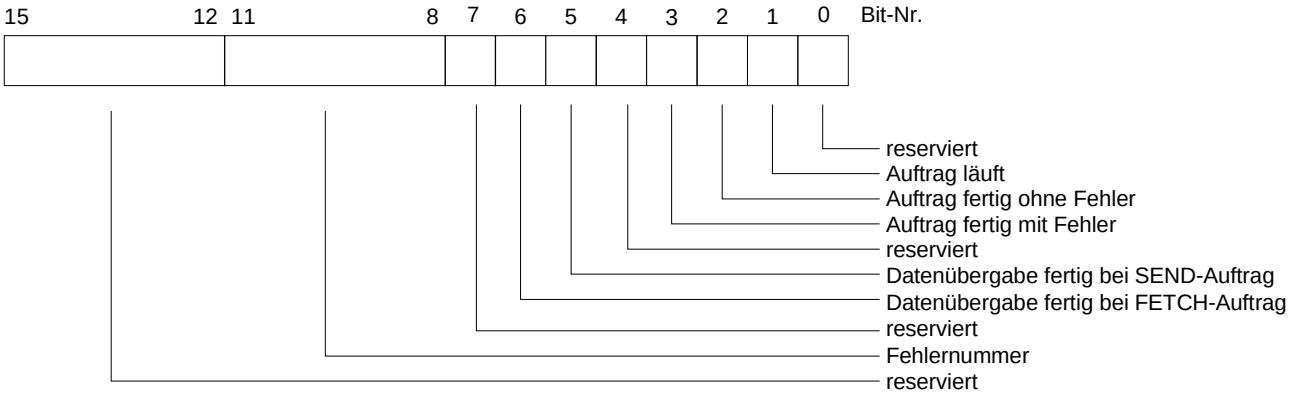
- Im Anzeigenwort ANZW
- im PAFE-Byte
- bei Aufruf des OB 35 in AKKU 1 mit einer Fehlernummer

Fehler bei der Bearbeitung eines Partner-Auftrags (Partner-SEND/Partner-FETCH) werden an den Partner mit einer Fehlernummer im Reaktions-telegramm REATEL gemeldet

Anzeigenwort (ANZW)

Das Anzeigenwort müssen Sie im Auftragsfach angeben. Es kann ein Merker, S-Merker oder Datenwort in einem DB oder erweiterten DB sein. Merker und S-Merker sind in der CPU immer vorhanden, Datenbausteine und erweiterte Datenbausteine müssen Sie zuerst anlegen.

Ins Anzeigenwort werden vom Systemprogramm Informationen über den **Status der Auftragsbearbeitung** eingetragen (z.B. "Auftrag läuft" oder "Auftrag fertig ohne/mit Fehler"). Wenn der Auftrag mit einem Fehler beendet wurde, steht im Bit 8 bis 11 zusätzlich eine Fehlernummer. Mit Hilfe dieser Fehlernummer können Sie den Fehler schneller lokalisieren und beseitigen.



Beschreibung der einzelnen Bits bzw. Bitgruppen, sofern sie nicht reserviert sind:

Bit 1 "Auftrag läuft"

- wird vom Systemprogramm gesetzt, wenn ein Auftrag gestartet (KBS-Bit 7 gesetzt) ist;
  Voraussetzung ist, dass das Auftragsfach richtig parametriert ist.
- wird vom Systemprogramm zurückgesetzt, wenn der Auftrag beendet ist.
- können Sie auswerten, wenn Sie wissen wollen, ob es sinnvoll ist einen neuen Auftrag zu starten.

Bit 2 "Auftrag fertig ohne Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag ohne Fehler abgeschlossen ist.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag mit dem gleichen ANZW das Bit 1 "Auftrag läuft" gesetzt wird.
- skönnen Sie auswerten, wenn Sie wissen wollen, ob der Auftrag ohne Fehler abgeschlossen wurde.

Bit 3 "Auftrag fertig mit Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag mit Fehler abgeschlossen wurde. In diesem Fall ist in Bit 8 bis 11 eine **Fehlernummmer** eingetragen.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm rückgesetzt, sobald für einen neuen Auftrag mit dem gleichen ANZW das Bit 1 "Auftrag läuft" gesetzt wird.
- können Sie auswerten, wenn Sie wissen wollen, ob der Auftrag mit Fehler abgeschlossen wurde. Aus Bit 8 bis 11 des Anzeigenwortes können Sie eine Fehlernummer entnehmen, welche die Ursache beschreibt.

Bit 5 "Datenübergabe fertig bei SEND-Auftrag"

- wird vom Systemprogramm gesetzt, wenn die Daten bei einem SEND-Auftrag vollständig aus der Datenquelle geholt wurden.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm rückgesetzt, sobald für einen neuen Auftrag, dem Sie ebenfalls dieses Anzeigenwort zugewiesen haben, Daten aus der Datenquelle geholt werden.
- können Sie auswerten, wenn Sie wissen wollen, ob die Daten bereits vollständig aus der Datenquelle geholt wurden. Sie können danach die Datenquelle wieder beschreiben, obwohl der SEND-Auftrag noch läuft.

Bit 6 "Datenübernahme fertig bei FETCH-Auftrag"

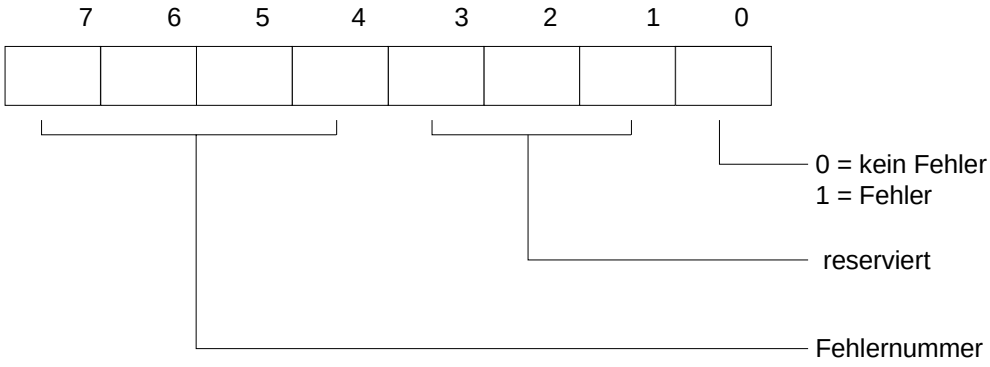
- wird vom Systemprogramm gesetzt, wenn die Daten bei einem FETCH-Auftrag vollständig ins Datenziel üertragen wurden.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag, dem Sie ebenfalls dieses Anzeigenwort zugewiesen haben, Daten ins Datenziel übertragen werden.
- können Sie auswerten, wenn Sie wissen wollen, ob die Daten bereits vollständig ins Datenziel übertragen wurden. Sie können danach die angeforderten Daten auswerten, obwohl der FETCH-Auftrag noch läuft.

Bit 8 bis 11 "Fehlernummer"

Ist Bit 3 = 1 ("Auftrag fertig mit Fehler") ist zusätzlich eine Fehlernummer eingetragen.

Das Parametrier-fehlerbyte (PAFE-Byte)

Das Parametrierfehlerbyte müssen Sie im Auftragsfach angeben. Es kann ein MB, S-MB oder DW (rechtes Datenbyte) in einem DB oder erweiterten DB sein.
Merker und S-Merker sind immer vorhanden, Datenbausteine und erweiterte Datenbausteine müssen Sie zuerst anlegen.
Das PAFE-Byte zeigt Fehler an, die unmittelbar mit der **Parametrierung im Auftragsfach** zusammenhängen.



Wenn im PAFE-Byte Bit 0 = 1, ist im Bit 4 bis 7 eine Fehlernummer hexadezimal eingetragen.

Datenaustausch

Der Datenaustausch zwischen den Partner findet auf drei Ebenen statt.

- Hardware-Ebene (Bit-Übertragungsschicht)
- Prozedur-Ebene (Sicherungssichicht)
- Telegramm-Ebene (Transportschicht)

Hardware-Ebene

Hardware-Ebene umfasst die physikalischen Bedingungen der Datenübertragung. In die CPU 928 können Sie verschiedenen Schnittstellenmodule stecken. (20 mA, RS 232C-Modul oder RS422).

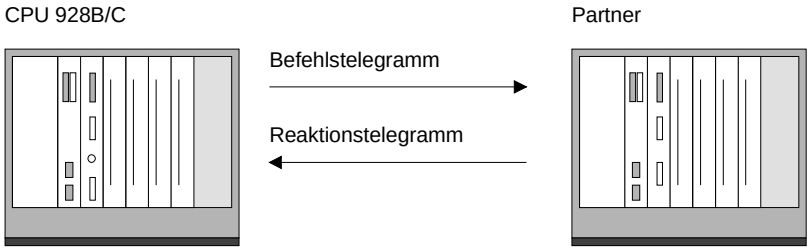
Prozedur-Ebene

Sämtliche Telegramme der RK 512 (Befehls- und Reaktionstelegramme) werden von der Prozedur-Ebene der Rechnerkopplung übertragen, d.h. mit der Prozedur 3964(R) gesendet oder empfangen. Die Prozedur verpackt die Telegramme mit einer Start- bzw. Endesequenz, versieht die entstehenden Datenblöcke mit einem Prüfzeichen (nur bei der Prozedur 3964R) und wickelt selbständig Telegrammwiederholungen ab bei fehlerhafter Telegrammquittung des Empfängers.

Telegramm-Ebene

Die Telegramm-Ebene entspricht der Transportschicht des ISO/OSI-Schichtenmodells.

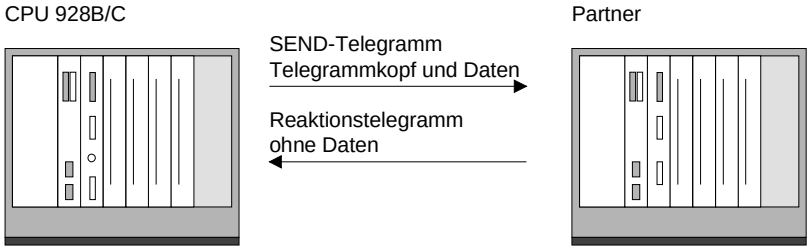
Die CPU 928 sendet ein Befehlstelegramm an den Partner. Dieser sendet ein Reaktionstelegramm an die CPU zurück.



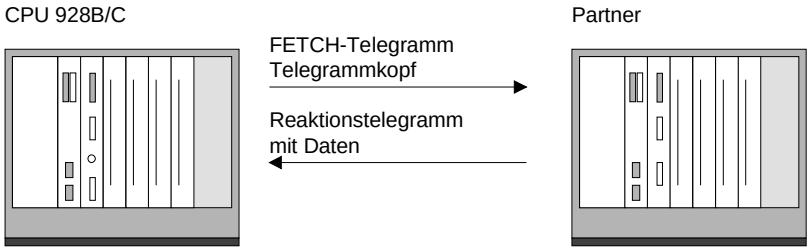
Befehlstelegramme sind entweder SEND-Telegramme (bei einem SEND-Auftrag) oder FETCH-Telegramme (bei einem FETCH-Auftrag).

Das SEND-Telegramm besteht aus einem Telegrammkopf und Daten.

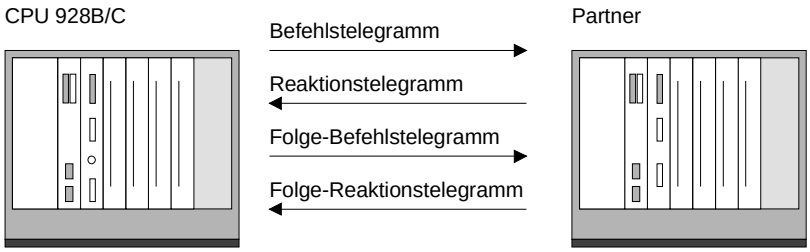
Bei einem SEND-Telegramm sendet der Partner ein Reaktionstelegramm ohne Daten.



Das FETCH-Telegramm besteht nur aus dem Telegrammkopf. Bei einem FETCH-Telegramm sendet der Partner ein Reaktionstelegramm mit Daten.



Überschreitet die Datenmenge 128 Byte, so werden bei allen Telegrammtypen Folgetelegramme gesendet.



Aufbau der Befehls-telegramme

Bei der RK 512 gibt es zwei Arten von Befehlstelegrammen, SEND-Telegramme und FETCH-Telegramme.

Das SEND-Telegramm besteht aus **Telegrammkopf und Daten**, das FETCH-Telegramm nur aus dem **Telegrammkopf**.

Der **Telegrammkopf** besteht aus **10 Byte**. Er enthält bei einem SEND-Telegramm Angaben über das Datenziel, beim FETCH-Telegramm Angaben über die Datenquelle.

Aufbau des Telegrammkopfes

Alle Werte sind in hexadezimaler Schreibweise dargestellt.

Byte	Inhalt	Bedeutung
1	00 oder FF	bei Befehlstelegrammen bei Folge-Befehlstelegrammen
2	00	
3	41 4F 45	"A" - für SEND-Auftrag mit Ziel DB oder "O" - für SEND-Auftrag mit Ziel DX oder "E" - für FETCH-Auftrag
4	44 58	zu übertragenden Daten sind aus DB zu übertragenden Daten sind aus DX
5	xx	DB/DX-Nr. **Datenziel** bei SEND-Auftrag, **Datenquelle** bei FETCH-Auftrag
6	xx	ab DW-Nr. **Datenziel** bei SEND-Auftrag, **Datenquelle** bei FETCH-Auftrag
7	xx	Länge High-byte **Länge** der zu übertragenden
8	xx	Länge Low-byte Daten in Wörter
9	xx	**Koordinierungsmerker** - Bytenummer, wenn Sie keinen Koordinierungsmerker angegeben haben, dann steht hier FFh.
10	xx	Bit 0 bis 3: **Koordinierungsmerker** -Bitnummer, wenn Sie keinen Koordinierungsmerker angegeben haben, dann steht hier Fh. Bit 4 bis 7: **CPU-Nummer** des Partners (Zahl von 1 bis 4) wenn Sie keine CPU-Nr., aber einen KM angegeben haben, dann steht hier 0h. wenn Sie keine CPU-Nr., aber keinen KM angegeben haben, dann steht hier Fh

Hinweis!

Der Telegrammkopf eines Folge-SEND- oder Folge-FETCH-Telegramms besteht nur aus Byte 1 bis 4.

Nachdem das Befehlstelegramm übertragen wurde, erwartet die RK 512 innerhalb einer Überwachungszeit ein Reaktionstelegramm des Partners.
Die Überwachungszeit ist abhängig von der Baudrate (bit/s).

ab	1200 bis 38400 Baud	5s
	600 Baud	7s
	300 Baud	10s
	150 Baud	15s
	110 Baud	10s

Die Baudrate legen Sie im statischen Parametersatz fest.

Aufbau des Reaktions-telegrammes

Das Reaktionstelegramm ohne Daten besteht aus **4 Byte** und enthält Informationen über den Verlauf des Auftrages.

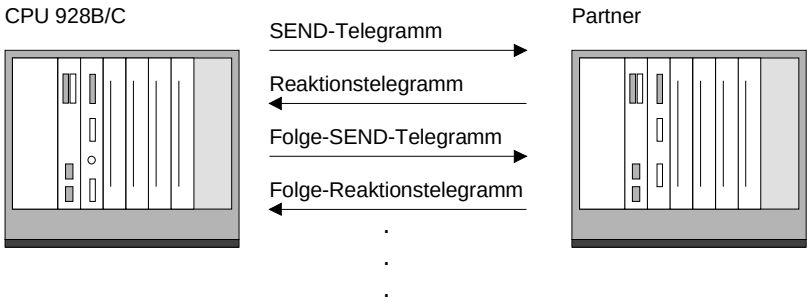
Alle Werte sind in Hexadezimaler Schreibweise dargestellt.

Byte	Inhalt	Bedeutung
1	00 oder FF	bei Befehlstelegrammen bei Folge-Befehlstelegrammen
2	00	
3	00	
4	00 oder xx	 Fehlernummer vom Partner

Das Reaktionstelegramm mit Daten (bei FETCH) enthält zusätzlich ab Byte 5 die Daten.

Telegrammbeschreibung

Detaillierter Ablauf eines SEND-Auftrages



SEND-Telegramme und Reaktionstelegramme laufen in folgender Reihenfolge ab:

Aktiver Partner:	Sendet ein SEND-Telegramm ab. Dieses enthält den Telegrammkopf und Daten.
Passiver Partner:	Empfängt das Telegramm, überprüft den Telegrammkopf und die Daten und quittiert mit einem Reaktionstelegramm.
Aktiver Partner:	Empfängt das Reaktionstelegramm. Wenn die Nutzdatenmenge **128 Byte** überschreitet, sendet er ein Folge-SEND-Telegramm. Dieses enthält Byte 1 bis 4 des Telegrammkopfes und weitere Daten.
Passiver Partner:	Empfängt das Folge-SEND-Telegramm, überprüft den Telegrammkopf und die Daten und quittiert mit einem Folge-Reaktions-telegramm.
.	.
.	.
.	.

Folge-SEND-Telegramme und Folge-Reaktionstelegramme werden gesendet, wenn die Nutzdatenmenge **128 Byte** überschreitet. Das erste Byte im Telegrammkopf von Folge-SEND- bzw. Folge-Reaktionstelegrammen ist zur Kennzeichnung jeweils FFh statt 00h.

Falls das SEND-Telegramm nicht fehlerfrei empfangen wurde, trägt der Partner eine Fehlernummer ins 4. Byte des Reaktionstelegramms ein.

Unten finden Sie den detailierten Ablauf eines SEND-Telegramms mit Reaktionstelegramm. Das SEND-Telegramm hat eine Nutzdatenlänge von höchstens 128 Byte und ist **störungsfrei** verlaufen (das 4.Byte im Reaktionstelegramm enthält keine Fehlernummer).

Ablauf eines störungsfreien SEND-Telegramms mit Reaktionstelegramm bei RK 512 mit der Prozedur 3964

	CPU 928			Partner	
Bedeutung		**Inhalt**			
SEND-Telegramm					
Startzeichen	———	STX	02h	⟶	
pos. Quittung	⟵	DLE	10h	———	
	———	1. Byte	00h	⟶	**Telegramm-kopf**
	———	2. Byte	00h	⟶	
SEND-Auftrag	———	3. Byte	41h	⟶	
Datenbaustein	———	4. Byte	44h	⟶	
Datenziel DB 10	———	5. Byte	0Ah	⟶	
DW 01	———	6. Byte	01h	⟶	
Länge	———	7. Byte	00h	⟶	
50 DW	———	8. Byte	32h	⟶	
kein KM	———	9. Byte	FFh	⟶	
alle CPUs	———	10.Byte	FFh	⟶	
1. Datenbyte	———	11. Byte		⟶	**Nutzdaten**
2. Datenbyte	———	12. Byte		⟶	
.	———	.		⟶	
.	———	.		⟶	
n. Datenbyte	———	x. Byte		⟶	
Endekennung	———	DLE	10h	⟶	
Endekennung	———	ETX	03h	⟶	
pos. Quittung	⟵	DLE	10h	———	
Reaktionstelegramm					
Startzeichen	⟵	STX	02h	———	
pos. Quittung	———	DLE	10h	⟶	
	⟵	1. Byte	00h	———	
	⟵	2. Byte	00h	———	
	⟵	3. Byte	00h	———	
Fehlernummer	⟵	4. Byte	00h	———	
Endekennung	⟵	DLE	10h	———	
Endekennung	⟵	ETX	03h	———	
pos. Quittung	———	DLE	10h	⟶	

Blockprüfsumme

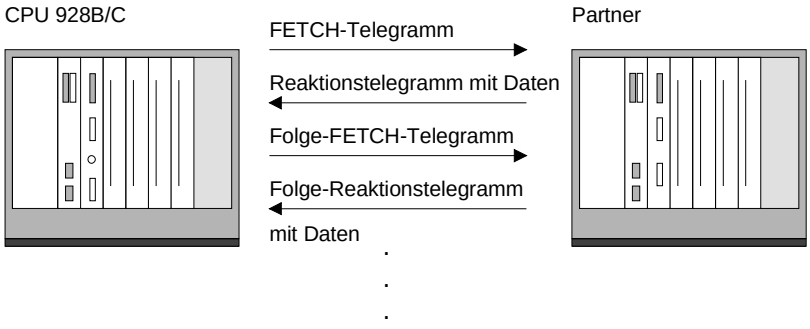
Beachten Sie, dass bei der Prozedur 3964R nach den Endekennungen DLE und ETX noch die Blockprüfsumme "BCC" (Längsparität) gesendet wird.

Das Folge-SEND-Telegramm hat eine Nutzdatenlänge von höchstens 128 Byte und ist störungsfrei verlaufen (das 4.Byte im Folge-Reaktionstelegramm enthält keine Fehlernummer).

Ablauf eines störungsfreien Folge-SEND-Telegramms mit Reaktionstelegramm bei RK 512 mit der Prozedur 3964

	CPU 928			**Partner**	
Bedeutung		**Inhalt**			
Folge-SEND-Telegramm					
Startzeichen	———	STX	02h	⟶	
pos. Quittung	⟵	DLE	10h	———	
Folgetelegramm	———	1. Byte	FFh	⟶	**Telegramm kopf**
	———	2. Byte	00h	⟶	
SEND-Auftrag	———	3. Byte	41h	⟶	
Datenbaustein	———	4. Byte	44h	⟶	
129. Datenbyte	———	5. Byte		⟶	**Nutzdaten**
130. Datenbyte	———	6. Byte		⟶	
.	———	.		⟶	
.	———	.		⟶	
n. Datenbyte	———	x. Byte		⟶	
Endekennung	———	DLE	10h	⟶	
Endekennung	———	ETX	03h	⟶	
pos. Quittung	⟵	DLE	10h	———	
Folge-Reaktionstelegramm					
Startzeichen	⟵	STX	02h	———	
pos. Quittung	———	DLE	10h	⟶	
Folge-Reaktions-	⟵	1. Byte	FFh	———	
telegramm	⟵	2. Byte	00h	———	
	⟵	3. Byte	00h	———	
Fehlernummer	⟵	4. Byte	00h	———	
Endekennung	⟵	DLE	10h	———	
Endekennung	⟵	ETX	03h	———	
pos. Quittung	———	DLE	10h	⟶	

Detaillierter Ablauf eines FETCH-Auftrages



FETCH-Telegramme und Reaktionstelegramme laufen in folgender Reihenfolge ab:

Aktiver Partner:	Sendet ein FETCH-Telegramm ab. Dieses enthält den Telegrammkopf.
Passiver Partner:	Empfängt das Telegramm, überprüft den Telegrammkopf und quittiert mit einem Reaktionstelegramm. Dieses enthält Daten.
Aktiver Partner:	Empfängt das Reaktionstelegramm. Wenn die Nutzdatenmenge **128 Byte** überschreitet, sendet er ein Folge-FETCH-Telegramm. Dieses enthält Byte 1 bis 4 des Telegrammkopfes.
Passiver Partner:	Empfängt das Folge-FETCH-Telegramm, überprüft den Telegrammkopf und quittiert mit einem Folge-Reaktionstelegramm. Dieses enthält weitere Daten.
.	.
.	.
.	.

Folge-FETCH-Telegramme und Folge-Reaktionstelegramme werden gesendet, wenn die Nutzdatenmenge **128 Byte** überschreitet. Das erste Byte im Telegrammkopf von Folge-FETCH- bzw. Folge-Reaktionstelegrammen ist zur Kennzeichnung jeweils FFh statt 00h.

Falls das FETCH-Telegramm nicht fehlerfrei empfangen wurde, trägt der Partner eine Fehlernummer ins 4. Byte des Reaktionstelegramms ein. Die angeforderten Daten werden dann nicht übertragen.

Auf der folgenden Seite finden Sie den detailierten Ablauf eines FETCH-Telegramms mit Reaktionstelegramm. Das Reaktionstelegramm beim FETCH hat eine Nutzdatenlänge von höchstens 128 Byte und ist **störungsfrei** verlaufen (das 4.Byte im Reaktionstelegramm enthält keine Fehlernummer).

Ablauf eines störungsfreien FETCH-Telegramms mit Reaktionstelegramm bei RK 512 mit der Prozedur 3964

	CPU 928			Partner	
Bedeutung		**Inhalt**			
FETCH-Telegramm					
Startzeichen	———	STX	02h	⟶	
pos. Quittung	⟵	DLE	10h	———	
	———	1. Byte	00h	⟶	**Telegramm-kopf**
	———	2. Byte	00h	⟶	
FETCH-Auftrag	———	3. Byte	45h	⟶	
Datenbaustein	———	4. Byte	44h	⟶	
Datenquelle DB 100	———	5. Byte	64h	⟶	
DW 100	———	6. Byte	64h	⟶	
Länge	———	7. Byte	00h	⟶	
50 DW	———	8. Byte	32h	⟶	
KM=M10.7	———	9. Byte	0Ah	⟶	
nur CPU 1	———	10.Byte	17h	⟶	
Endekennung	———	DLE	10h	⟶	
Endekennung	———	ETX	03h	⟶	
pos. Quittung	⟵	DLE	10h	———	
Reaktionstelegramm mit Daten					
Startzeichen	⟵	STX	02h	———	
pos. Quittung	———	DLE	10h	⟶	
	⟵	1. Byte	00h	———	
	⟵	2. Byte	00h	———	
	⟵	3. Byte	00h	———	
Fehlernummer	⟵	4. Byte	00h	———	
1. Datenbyte	⟵	5. Byte		———	**Nutzdaten**
2. Datenbyte	⟵	6. Byte		———	
.	⟵	.		———	
.	⟵	.		———	
n. Datenbyte	⟵	x. Byte		———	
Endekennung	⟵	DLE	10h	———	
Endekennung	⟵	ETX	03h	———	
pos. Quittung	———	DLE	10h	⟶	

Blockprüfsumme

Beachten Sie, dass bei der Prozedur 3964R nach den Endekennungen DLE und ETX noch die Blockprüfsumme "BCC" (Längsparität) gesendet wird.

Das Folge-FETCH-Telegramm hat eine Nutzdatenlänge von höchstens 128 Byte und ist störungsfrei verlaufen (das 4.Byte im Folge-Reaktionstelegramm enthält keine Fehlernummer).

Ablauf eines störungsfreien Folge-FETCH-Telegramms mit Folge-Reaktionstelegramm bei RK 512 mit der Prozedur 3964

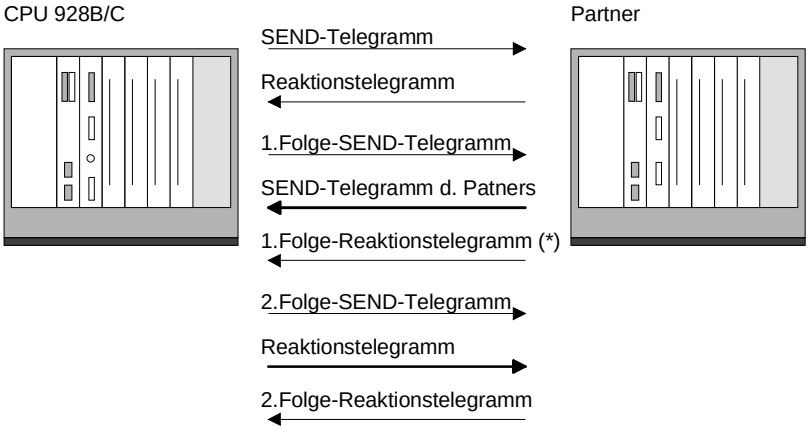
	CPU 928			Partner	
Bedeutung		**Inhalt**			
Folge-FETCH-Telegramm					
Startzeichen	———	STX	02h	→	
pos. Quittung	←	DLE	10h	———	
Folgetelegramm	———	1. Byte	FFh	→	**Telegramm-kopf**
	———	2. Byte	00h	→	
FETCH-Auftrag	———	3. Byte	45h	→	
Datenbaustein	———	4. Byte	44h	→	
Endekennung	———	DLE	10h	→	
Endekennung	———	ETX	03h	→	
pos. Quittung	←	DLE	10h	———	
Folge-Reaktionstelegramm					
Startzeichen	←	STX	02h	———	
pos. Quittung	———	DLE	10h	→	
Folge-Reaktions-	←	1. Byte	FFh	———	
telegramm	←	2. Byte	00h	———	
	←	3. Byte	00h	———	
Fehlernummer	←	4. Byte	00h	———	
129. Datenbyte	←	5. Byte		———	**Nutzdaten**
130. Datenbyte	←	6. Byte		———	
.	←	.		———	
.	←	.s		———	
n. Datenbyte	←	x. Byte		———	
Endekennung	←	DLE	10h	———	
Endekennung	←	ETX	03h	———	
pos. Quittung	———	DLE	10h	→	

Quasi-Vollduplex-betrieb

Quasi-Vollduplex bedeutet: die Partner können zu beliebigen Zeitpunkten Telegramme senden, außer wenn der Partner gerade sendet.

Falls beide Partner gleichzeitig senden wollen, kann beispielsweise vor dem Reaktionstelegramm des Partners ein Befehlstelegramm eintreffen.

Im folgenden Beispiel wird das erste Folge-Reaktionstelegramm (*) des Partners zum ersten Folge-SEND-Telegramm erst nach dem SEND-Telegramm des Partners gesendet.



Teil 15 Datenübertragung 3964(R)

Überblick

Die zweite Schnittstelle SI 2 der CPU 928 können Sie nutzen als Schnittstelle für die Punkt-zu-Punkt-Kommunikation mit den Prozeduren 3964 oder 3964R.

Die Prozeduren 3964 bzw. 3964R der CPU 928 verwendet den identischen Telegrammaufbau wie die Kommunikationsprozessoren SSM BG4xD

Nachfolgend sind beschrieben:

- Einführung - Einbindung ins Anwenderprogramm
- Parametrierungen - statischer Parametersatz
- Betrieb - SEND-, Partner-SEND-Aufträge
- Prozeduren - Beispiele Senden, Empfangen.

Inhalt

Thema **Seite**

Teil 15 Datenübertragung 3964(R) 15-1

Einführung 3964 und 3964R 15-2

Parametrierung 15-9

Inbetriebnahme 15-22

Betrieb 15-24

Prozeduren 15-30

Einführung 3964 und 3964R

Mit den Prozedur 3964 / 3964R können zwischen zwei Punkt-zu Punkt gekoppelten Kommunikationspartnern Daten ausgetauscht werden.

*Zur Vereinfachung werden in diesem Kapitel die Prozeduren 3964 und 3964R zu einem Ausdruck **3964(R)** zusammengefasst sofern die Beschreibung sich auf beide Prozeduren bezieht.*

Die Datenübertragung mit der Prozedur 3964(R) gewährleistet größtmögliche Übertragungssicherheit, da die Übertragung mit dieser Prozedur folgende Schichten des ISO/OSI-Schichtenmodelles (ISO IS 7498) umfasst:

- Bit-Übertragungsschicht (Schicht 1):
  In dieser Schicht wird die physikalische Übertragung der Datenbytes festgelegt (Physik, Baudrate,...).

- Sicherungsschicht (Schicht 2):
  Die Prozedur 3964(R) fügt dem Datenstrom Start- und Endezeichen hinzu. Weiterhin ist sie in der Lage, den Verlust bzw. die fehlerhafte Übertragung von Datenblöcken zu erkennen und mit Wiederholungen darauf zu reagieren.

Das Protokoll der Prozedur gewährleistet lediglich eine gesicherte Übertragung der Daten, nicht jedoch die Verarbeitung bzw. Auswertung der übertragenen Daten.

Hinweis!

Die Datenübertragung mit der Prozedur 3964(R) arbeitet mit den gleichen Prozeduren wie die Rechnerkopplung mit RK 512.

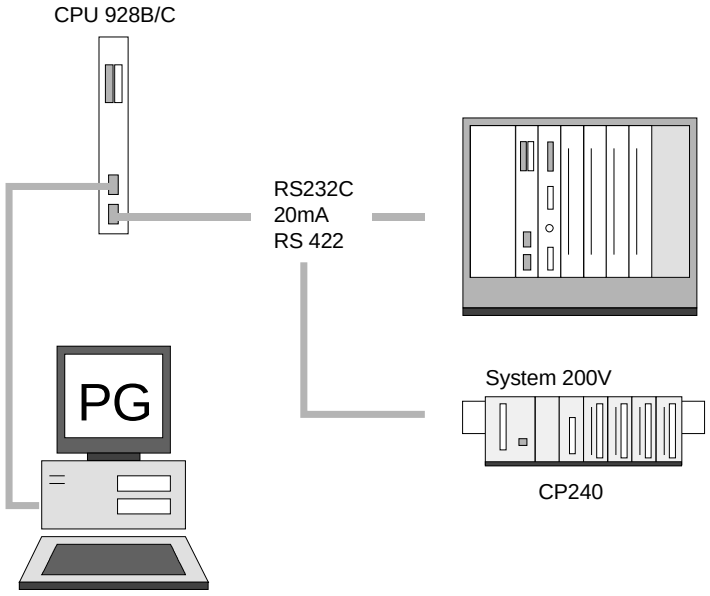
Wird ein Telegramm vom Partner positiv quittiert (angenommen/übernommen), ist das Telegramm korrekt empfangen worden. Das heißt nicht, dass das Telegramm vom Empfänger bearbeitet wird (z.B. können die Daten nicht vom Systemprogramm ins Empfangsfach geschrieben werden, wenn dieses zu kurz ist).

Wenn Sie sicherstellen wollen, dass das Telegramm auch bearbeitet wird, müssen Sie entweder die Rechnerkopplung RK 512 benutzen oder die Datensicherung mit den Nutzdaten selbst vornehmen

Anwendungs-bereich

Die Datenübertragung mit den Prozeduren 3964 und 3964(R) realisiert den Datenaustausch zwischen der CPU 928 in einem AG 135U bzw. AG 155U und einem der folgenden Partner:

- eine weitere **CPU 928** in den AGs 135U und 155U
- ein **CP 240** des System 200V
- **BG41, BG42, BG43** CP-Baugruppe mit Kommunikationsprozessor in den Automatisierungsgeräten 115U, 115H, 135U, 155U und 155H.
- **BG4xD** CP-Baugruppe AG 115U, 115H, 135U, 150U, 155U und 155H
- ein **Fremdrechner**, der mit derselben Prozedur arbeitet.



Für die Datenübertragung mit der Prozedur 3964(R) stehen Ihnen verschiedene Schnittstellenmodule zur Verfügung:

- 20 mA-Modul
- RS 232-Modul
- RS 422-Modul.

Die Verbindung zum Partner stellen Sie mit einer Standard-Steckleitung her. Die Übertragungsgeschwindigkeit für den Datenaustausch stellen Sie entsprechend dem jeweiligen Koppelpartner ein.

Zugelassen sind:

- 110 Bit/s bis 38 400 Bit/s bei RS232C-Übertragung
- 110 Bit/s bis 38 400 Bit/s bei 20mA-Übertragung
- 110 Bits bis 38 400 Bit/s bei RS 422-Übertragung.

Aufträge

Mit der Datenübertragung mit der Prozedur 3964(R) können Sie **aktiv**:

- Daten senden : **SEND**-Auftrag
  Die zu sendenden Daten stehen im Sendefach.

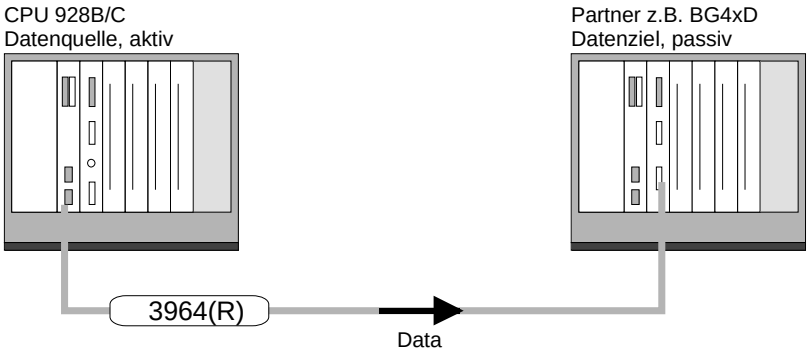
Mit der Datenübertragung mit der Prozedur 3964(R) können Sie **passiv**:

- Daten empfangen: Partner-**SEND**-Auftrag
  Die empfangenen Daten stehen im Empfangsfach.

SEND-Auftrag und Partner-SEND-Auftrag bilden ein Auftragspaar. Ein SEND-Auftrag bei der CPU 928 führt zu einem Partner-SEND-Auftrag beim Partner und umgekehrt.

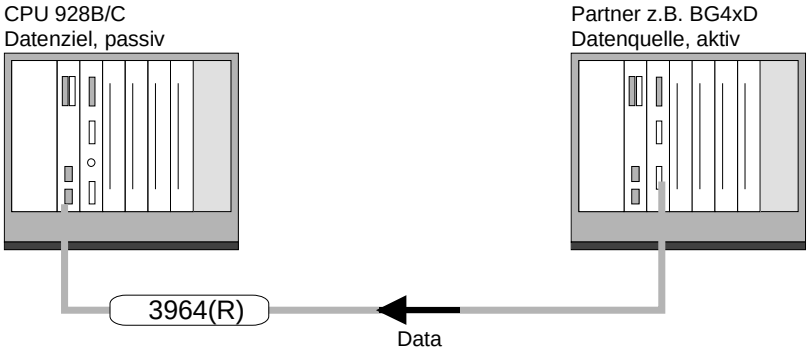
SEND-Auftrag

Die CPU 928B/C ist aktiv, d.h. sie sendet Daten zum Partner. Der Partner ist passiv - er empfängt Daten.



Partner-SEND-Auftrag

Der Partner ist aktiv, d.h. er sendet Daten zur CPU 928B/C. Die CPU 928B/C ist passiv - sie empfängt Daten.



Einbindung ins Anwender-programm

In diesem Abschnitt finden Sie eine Übersicht, wie die Datenübertragung mit der Prozedur 3964(R) in ein Anwenderprogramm eingebunden wird.

Für das Erstellen der Anwenderprogramme wird Ihnen folgende Vorgehensweise empfohlen:

Vorabüberlegungen

- Mit welchem Partner soll die CPU Daten austauschen?
- Welches Schnittstellenmodul setzen Sie ein?
- Welche Daten sollen ausgetauscht werden?
- Wie sollen die Aufträge lauten?

Parametrierung der Schnittstelle

Zur Parametrierung der Schnittstelle benötigen Sie:

- DX 2
- statischen Parametersatz
- Sendefach
- Empfangsfach
- Koordinierungsbyte Senden (KBS)
- Koordinierungsbyte Empfangen (KBE).

DX 2

In den erweiterten Datenbaustein DX 2 müssen Sie folgende Informationen eintragen:

- Kopplungstyp: Datenübertragung mit der Prozedur 3964 oder 3964R
- Lage des statischen Parametersatzes
- Lage des Sendefachs
- Länge des Sendefachs
- Lage des Empfangsfachs
- Länge des Empfangsfachs
- Lage des Koordinierungsbytes Senden (KBS) und Empfangen (KBE).

statischer Parametersatz

In den DB/DX mit dem statischen Parametersatz sind die Parameter zur Bit-Übertragungsschicht und zur Sicherungsschicht einzutragen.

Parameter der Bit-Übertragungsschicht (Schicht 1):

- Baudrate (Bit/S)
- Parität
- Bit pro Zeichen
- Stoppbits.

Parameter der Sicherungsschicht (Schicht 2)

In der Sicherungsschicht werden die Eigenschaften der Übertragungsprozedur 3964(R) festlegen:

- Modus: Prozedur 3964(R) mit Standardwerten oder parametrierbar
- Priorität
- Zeichenverzugszeit
- Quittungsverzugszeit
- Aufbauversuche
- Wiederholanzahl

Sendefach

Im DB/DX mit dem Sendefach stehen die zu sendenden Daten. Das Sendefach darf maximal 2049 Wörter lang sein.

Empfangsfach

Im DB/DX mit dem Empfangsfach stehen die zu empfangenen Daten. Das Empfangsfach darf maximal 2049 Wörter lang sein.

KBS

Mit dem Koordinierungsbyte Senden (KBS) können Sie:

- einen SEND-Auftrag starten
- die Datenübertragung mit der Prozedur 3964(R) zurücksetzen.

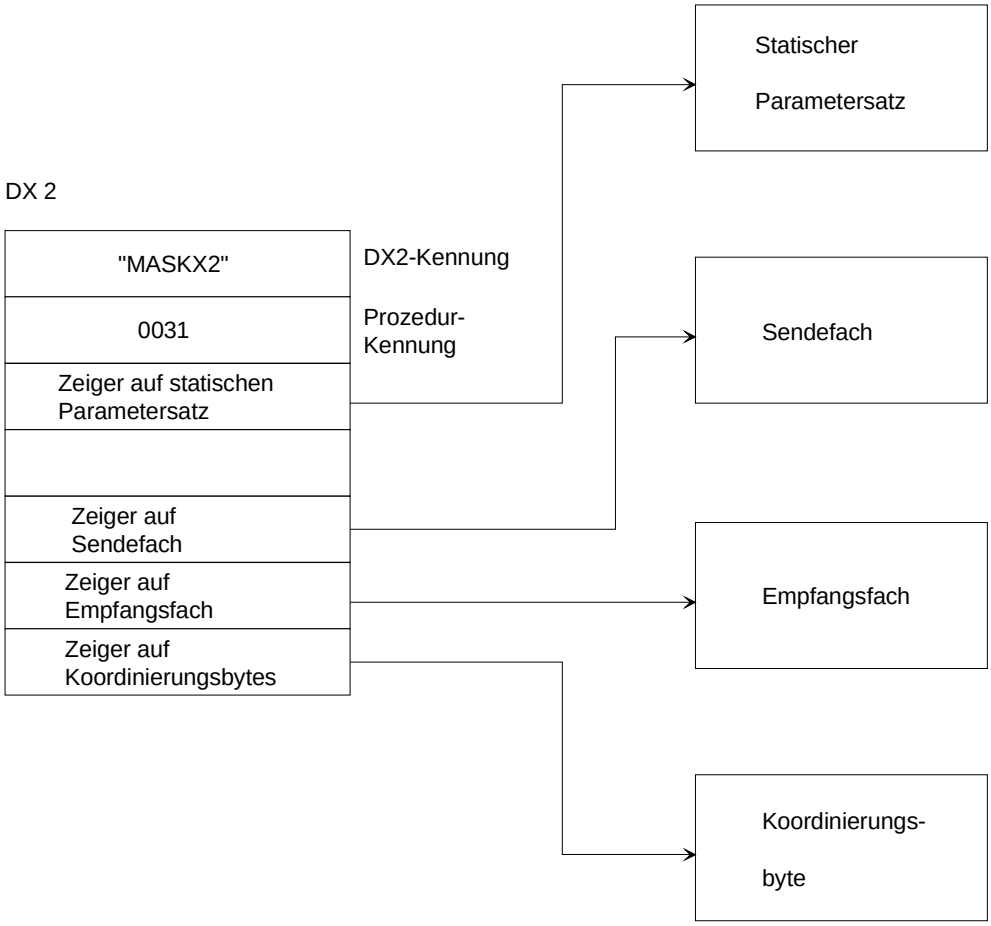
KBE

Mit dem Koordinierungsbyte Empfangen (KBE) können Sie:

- das Empfangsfach freigeben und sperren.

Den Parametersatz im DX 2, den statischen Parametersatz und die dynamischen Parametersätze können Sie mit dem Programmiertool MC5 von VIPA erstellen.

Komponenten des Anwender-programms



CPU aktiv betreiben

Bei einer ausschließlich **aktiv** betriebenen CPU, die selbst keine Daten empfängt, benötigen Sie für die Datenübertragung mit der Prozedur 3964(R) kein Empfangsfach.

CPU passiv betreiben

Bei einer ausschließlich **passiv** betriebenen CPU, die selbst keine Aufträge anstößt, benötigen Sie für die Datenübertragung mit der Prozedur 3964(R) kein Sendefach.

Hinweis!

Ist in der CPU kein DX 2 vorhanden, wird die zweite Schnittstelle als PG-Schnittstelle initialisiert.

Hinweis!

Die Parameter im DX 2 und im statischen Parametersatz legen Sie fest, bevor Sie die Datenübertragung mit der Prozedur 3964(R) in Betrieb nehmen.

Die Parameter im DX 2 werden nur bei Neustart vom Systemprogramm übernommen.

Die Parameter im statischen Parametersatz (DW 0 bis DW 4) werden nur bei Spannungswiederkehr und bei Neustart vom Systemprogramm übernommen.

Die Parameter im statischen Parametersatz ab DW 5 werden nur bei Spannungswiederkehr, Neustart und dem Auftrag "Datenübertragung mit der Prozedur 3964(R) rücksetzen" vom Systemprogramm übernommen.

Den Inhalt des Sendefachs und des KBS dürfen Sie nur verändern, wenn kein SEND-Auftrag läuft.

Den Inhalt des Empfangsfachs und des KBE dürfen Sie nur verändern, wenn kein Partner-SEND-Auftrag läuft.

Detaillierte Informationen mit welchen Werten Sie die Datenübertragung mit den Prozeduren 3964(R) parametrieren können, stehen in den nachfolgenden Unterkapiteln.

Sie haben zwei Möglichkeiten

- Sie benutzen das Programmiertool MC5 von VIPA.
- Sie parametrieren direkt in STEP® 5 von Siemens über Eingabe in Datenbausteine

Parametrierung

Für die Parametrierung können Sie das Programmiertool MC5 von VIPA einsetzen. Hiermit können Sie alle hier aufgeführten Bausteine parametrieren.

DX 2 parametrieren

Im DX 2 wird der Kopplungstyp vermerkt. Außerdem enthält er die Zeiger auf sämtliche benötigte Parameter und Parametersätze.

Die Parametrierung im DX 2 beginnt absolut bei DW 0. Alle Zahlenwerte sind in hexadezimaler Schreibweise dargestellt.

DW	Parameter	Bedeutung
0	4D41	MA
1	534B	SK
2	5832	X2
Kopplungstyp		
3	0031	Datenübertragung mit 3964(R)
Zeiger auf statischen Parametersatz		
4	44xx oder 58xx	DB-Nr.xx oder [4] DX-Nr.xx
5	xxxx	ab DW-Nr. xxxx
6	0000	reserviert
7	0000	reserviert
Zeiger auf Sendefach [1]		
8	44xx oder 58xx	DB-Nr. xx [3] DX-Nr.xx
9	xxxx	ab DW-Nr. xxxx
10	xxxx	Länge des Sendefachs in Wörter (Wertebereich 0002h bis 0801h)
Zeiger auf Empfangsfach [2]		
11	44xx oder 58xx	DB-Nr.xx oder [4] DX-Nr.xx
12	xxxx	ab DW-Nr.xxxx
13	xxxx	Länge des Empfangsfachs in Wörter (Wertebereich 0002h bis 0801h)
Zeiger auf KBS / KBE [3]		
14	4D00 oder 5300 oder 44xx oder 58xx	Merker oder [4] S-Merker oder DB-Nr. xx oder DX-Nr. xx
15	xxxx	MW-Nr., SW-Nr. oder DW-Nr.xxxx

[1] Nicht notwendig, wenn CPU nur passiv ist. Tragen Sie dann in DW 8 bis DW 10 den Wert 0000 ein.

[2] Nicht notwendig, wenn CPU nur aktiv ist. Tragen Sie dann in DW 11 bis DW 13 den Wert 0000 ein.

[3] Die Lage von KBS und KBE können Sie nur gemeinsam festlegen.

[4] Wertebereich für xx von 3 bis FFh (3-255)

Beispiel:

In diesem Beispiel sehen Sie, wie Sie einen DX 2 mit folgenden Grunddaten parametrieren.

- Der statische Parametersatz liegt in DB 10 ab DW 4.
- Das Sendefach liegt im DB 6 ab DW 2, Länge 10 Wörter.
- Das Empfangsfach liegt in DB 8 ab DW 1, Länge 6 Wörter.
- Das Koordinierungsbyte Senden (KBS) und Empfangen (KBE) liegen in MW 6.

Dies ergibt folgende Parametrierung:

DX 2

```
0:      KH=4D41;    MA
1:      KH=534B;    SK
2:      KH=5832;    X2
3:      KH=0031;    Kopplungstyp Prozedur 3964(R)
4:      KH=440A;    Statischer Parametersatz in DB 10
5       KH=0004;    Ab DW 4
6       KH=0000;    reserviert
7       KH=0000;    reserviert
8:      KH=4406;    Sendefach in DB 6
9       KH=0002;    Ab DW 2
10      KH=000A;    Länge 10 Wörter
11      KH=4408;    Empfangsfach in DB 8
12      KH=0001;    ab DW 1
13      KH=0006;    Länge 6 Wörter
14      KH=4D00;    Koordinierungsbytes im Merkerbereich
15      KH=0006;    KBS-Merkerbyte 6, KBE-Merkerbyte 7
16      KH=xxxx     gehört nicht mehr zur Parametrierung des
                    DX 2, beliebiger Inhalt, muss nicht
                    vorhanden sein.
```

Statischer Parametersatz

Im statischen Parametersatz tragen Sie die Parameter zur Bit-Übertragungsschicht und zur Sicherungsschicht ein.

In der nachfolgenden Tabelle steht, mit welchen Werten Sie den statischen Parametersatz in einem Datenbaustein oder erweiterten Datenbaustein hinterlegen können.

Die Parametrierung beginnt bei dem im DX 2 parametrierten DW.

Alle Zahlenwerte sind hexadezimal dargestellt.

DW (relativ)	Werte-Bereich	Bedeutung
Baudrate		
0	0001 0002 0003 0004 0005 0006 0007 0008 0009 000A	110 Baud 150 Baud 300 Baud 600 Baud 1200 Baud 2400 Baud 4800 Baud 9600 Baud 19200 Baud 38400 Baud
Parität		
1	0000 0001 0002	keine (no) ungerade (odd) gerade (even)
Bit pro Zeichen		
2	0006 0007 0008	6 Bit pro Zeichen 7 Bit pro Zeichen 8 Bit pro Zeichen
Stoppbits		
3	0001 0002 0003	1 Stoppbit 1 + ½ Stoppbits 2 Stoppbits
4	0000	reserviert
Modus (siehe Hinweis)		
5	0001 0002 0003 0004	Prozedur 3964 mit Standardwerten Prozedur 3964R mit Standardwerten Prozedur 3964 parametrierbar Prozedur 3964R parametrierbar
Priorität		
6	0000 0001	niedrig hoch

...Fortsetzung

...Fortsetzung

DW (relativ)	Werte-Bereich	Bedeutung
Zeichenverzugszeit (siehe Hinweis)		
7	0001 bis FFFF	Überwachungszeit 0,01 s bis 655,35 s (Einheit: 10ms)
Quittungsverzug (siehe Hinweis)		
8	0001 bis FFFF	Überwachungszeit 0,01 s bis 655,35 s (Einheit: 10ms)
Aufbauversuche(siehe Hinweis)		
9	0001 bis 00FF	Anzahl
Wiederholanzahl (siehe Hinweis)		
10	0001 bis 00FF	Anzahl

Hinweis!

Die Datenwörter 7 bis 10 werden nur dann ausgewertet, wenn Sie als Modus 0003 oder 0004 eingetragen haben.

Wenn Sie als Modus 0001 oder 0002 eingetragen haben, werden die Standardwerte verwendet. Beachten Sie aber, dass die Datenwörter 7 bis 10 trotzdem vorhanden sein müssen.

Die Standardwerte für Modus 0001 und Modus 2 sind:

	Modus 0001	Modus 0002
Zeichenverzugzeit	220 ms	220 ms
Quittungsverzugszeit	550ms	2000 ms
Aufbauversuche	6	6
Wiederholanzahl	6	6

Bedeutung der Parameter

Baudrate

Geschwindigkeit der Datenübertragung, Angabe in Bits/s (Baud).

Parität

Anzahl der Bits mit dem Wert "1" in einer einheitlichen langen Folge von Informationsbits (z.B. ein Byte). Die Parität ist, je nach Wert, "gerade" oder "ungerade". Zur Paritätskontrolle wird diese Folge von Informationsbits um ein weiteres Bit, das Paritätsbit (parity bit), erweitert, das durch seinen addierten Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt.

Bit pro Zeichen

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

Stoppbits

Dauer der Stoppbits relativ zur Übertragungsdauer eines Informationsbits. Die Stoppbits werden bei der Start-Stopp-Übertragung jedem zu übertragenden Zeichen nachgesetzt.

Modus

Variante der möglichen Übertragungsprozeduren, 3964(R), jeweils mit Standardwerten oder parametrierbar.

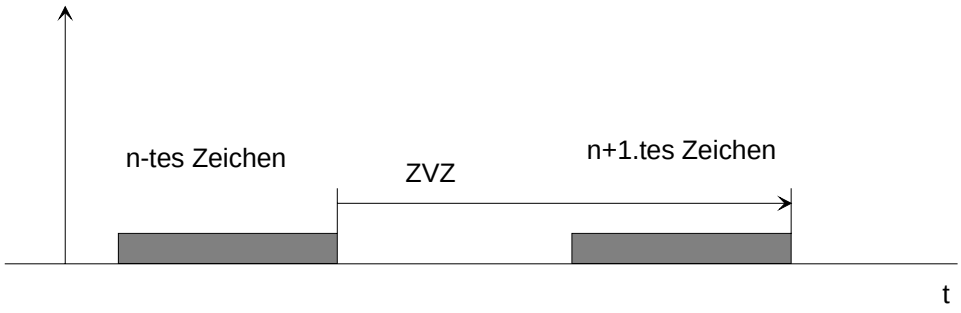
Priorität

Ein Partner hat Priorität, wenn sein Sendewunsch Vorrang vor dem Sendewunsch des anderen Partners hat.

Ein Partner hat niedere Priorität, wenn sein Sendewunsch hinter den Sendewunsch des anderen Partners zurückstehen muss. Beim Betrieb der Rechnerkopplung müssen Sie beide Koppelpartner mit verschiedener Priorität parametrieren, d.h. **A** bekommt Priorität =**1** (hoch), **B** bekommt Priorität = **0** (niedrig).

Zeichenverzugszeit (ZVZ)

Der maximal zulässige zeitliche Abstand zwischen zwei empfangenen Zeichen (siehe folgende Tabelle).



Quittungsverzugzeit (QVZ)

Die maximal zulässige Zeitspanne bis zur Quittung des Partners bei Verbindungsaufbau/-abbau.

Baudrate	kleinste zulässige ZVZ	kleinste zulässige QVZ
110 Baud	120 ms	120 ms
150 Baud	100 ms	100 ms
300 Baud	60 ms	60 ms
600 Baud	40 ms	40 ms
1200 Baud	30 ms	30 ms
2400 Baud	20 ms	20 ms
4800 Baud	20 ms	20 ms
9600 Baud	20 ms	20 ms
9600 Baud	20 ms	20 ms
19600 Baud	20 ms	20 ms
38400 Baud	20 ms	20 ms

Aufbauversuche

Maximale Anzahl der Versuche der CPU 928 eine Verbindung aufzubauen.

Wiederholanzahl

Maximale Anzahl der Telegrammwiederholungen (inklusive erstes Telegramm) bei Fehlern.

Beispiel

In diesem Beispiel sehen Sie, wie Sie einen statischen Parametersatz im DB 10 ab DW 4 parametrieren.

DB10

```
0:     KH =xxxx;     gehört nicht zum statischen
                     Paramertersatz, beliebiger Inhalt
1:     KH =xxxx;
2:     KH =xxxx;
3:     KH =xxxx;
4:     KH =0008;     Baudrate = 9600 Baud
5:     KH =0002;     Parität = 2 = Gerade
6:     KH =0007;     Bit pro Zeichen =7
7:     KH =0001;     Stoppbits = 1
8:     KH =0000;     reserviert
9:     KH =0003;     Modus = 3 = Prozedur 3964 parametrier-
                     bar
10:    KH =0001;     Priorität = 1 = hoch
11:    KH =00FF;     Zeichenverzugszeit = 255 x 0,01 s
                     (= 2,55 s)
12:    KH =010A;     Quittungsverzugzeit =266 x 0,01 s
                     (= 2,66 s)
13:    KH =0003;     Aufbauversuche = 3
14:    KH =0005;     Wiederholanzahl =5
15:    KH =xxxx;     gehört nicht mehr zum statischen Para-
                     metersatz, beliebiger Inhalt, muss
                     nicht vorhanden sein.
16:
  :
```

Bei der Datenübertragung mit der Prozedur 3964(R) müssen die Parameter der Bit-Übertragungsschicht beim Partner mit denen der CPU 928 übereinstimmen.

Die Parameter der Sicherungsschicht müssen beim Partner wie folgt gesetzt werden:

- Verwendung der gleichen Prozedur wie bei der CPU 928
- entgegengesetzte Priorität wie bei der CPU 928.

Sendefach

Das Sendefach liegt in einem Datenbaustein (DB) oder erweiterten Datenbaustein (DX).

Das Sendefach enthält die zu sendenden Daten sowie im ersten Wort die Längenangabe (Angabe in Byte) der zu übertragenden Daten. Sie können so für jeden Auftrag selbst festlegen, wie viele Daten an den Koppelpartner gesendet werden. Die Längenangabe schließt das Wort mit der Längenangabe selbst nicht mit ein. Die Längenangabe wird nicht mitgesendet.

Die Länge des Sendefachs beträgt maximal 2049 Wörter (4098 Byte). Sie legen sie (Angabe in Wörter) im DX 2 fest. Achten Sie darauf, dass die Längenangabe der zu übertragenden Daten (Angabe in Byte) im ersten Wort des Sendefachs nicht größer ist als die im DX 2 parametrierte Länge (Angabe in Wörter) des Sendefachs minus dem ersten Wort mit der Längenangabe. Parametrieren Sie die Länge des Sendefaches entsprechend Ihrer aktuellen Aufgabe.

Beispiel: Sendefach

Das Beispiel zeigt, wie Sie ein Sendefach im Datenbaustein 6 ab Datenwort 2 mit einer Länge von 10 Wörtern parametrieren können. Es sollen 7 Byte gesendet werden.

DB 6

```
0:     KH =xxxx;     gehört nicht zum Sendefach, beliebiger
                     Inhalt
1:     KH =xxxx;
2:     KH =0007;     Längenangabe in Byte, Sendefachbeginn
3:     KH =0102;     zu sendende Daten: 0102
4:     KH =0304;     zu sendende Daten: 0304
5:     KH =0506;     zu sendende Daten: 0506
6:     KH =07yy;     zu sendende Daten: 07, yy wird nicht
                     mehr gesendet
7:     KH =yyyy;     Inhalt des Sendefachs, wird hier nicht
                     gesendet
8:     KH =yyyy;
9:     KH =yyyy;
10:    KH =yyyy;
11:    KH =yyyy;
12:    KH =xxxx;     gehört nicht zum Sendefach, beliebiger
                     Inhalt, muss nicht vorhanden sein
:
```

Empfangsfach

Das Empfangsfach liegt im einem DB oder erweiterten DB (DX).

Das Empfangsfach enthält die empfangenen Daten, sowie im ersten Wort die Längenangabe (Angabe in Byte) der empfangenen Daten. Diese wird vom Systemprogramm eingetragen, damit Sie erkennen können, wieviele Daten empfangen wurden. Die Längenangabe schließt das Wort mit der Längenangabe selbst nicht ein.

Die Länge des Empfangsfaches beträgt maximal 2049 Wörter (4098 Byte). Sie legen sie (Angabe in Wörter) im DX 2 fest. Es können nur so viele Byte empfangen werden, wie Sie im DX 2 als Länge (Angabe in Wörter) des Empfangsfaches angegeben haben minus dem ersten Wort mit der Längenangabe im Empfangsfach.

Beispiel: Empfangsfach

Das Beispiel zeigt, wie ein Empfangsfach im DB 8 ab DW 1 mit einer Länge von 6 Wörtern aufgebaut ist. Es wurden 7 Byte empfangen.

DB8

```
0:     KH =xxxx;     gehört nicht zum Empfangsfach,
                     beliebiger Inhalt
1:     KH =0007;     Längenangabe in Byte, Empfangsfach-
                     beginn
2:     KH =0102;     empfangene Daten: 0102
3:     KH =0304;     empfangene Daten: 0304
4:     KH =0506;     empfangene Daten: 0506
5:     KH =07yy;     empfangene Daten: 07,yy wurde nicht
                     mehr empfangen
6:     KH =yyyy;     Inhalt Empfangsfach, wurde nicht
                     empfangen
7:     KH =xxxx;     gehört nicht zum Empfangsfach,
                     beliebiger Inhalt, muss nicht
                     vorhanden sein
:
```

Koordinierungsbytes Senden und Empfangen (KBS, KBE)

Mit dem KBS können Sie folgende Aufträge starten:

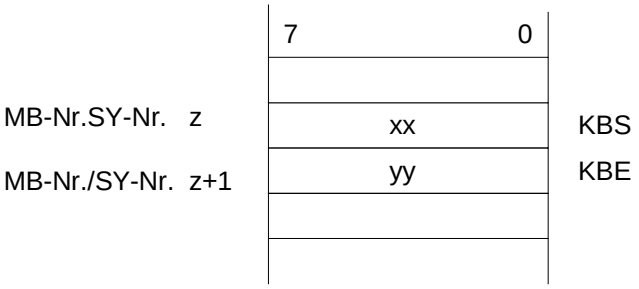
- **SEND**-Auftrag
- die Datenübertragung mit der Prozedur 3964(R) rücksetzen.

Mit dem KBE können Sie:

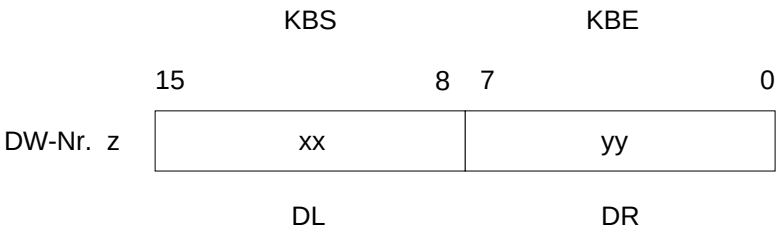
- das Empfangsfach freigeben und sperren.

Die Koordinierungsbytes können Merker bzw. S-Merker sein oder im einem Datenbaustein bzw. erweiterten Datenbaustein liegen. Merker und S-Merker sind in der CPU vorhanden, Datenbausteine und erweiterte Datenbausteine müssen Sie anlegen.

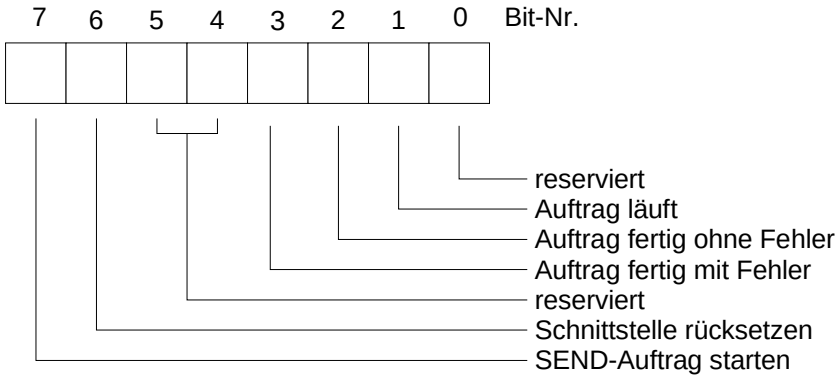
Darstellung im Merker- bzw. S-Merkerbereich:



Darstellung im DB bzw. im erweiterten DB:



KBS Das Koordinierungsbyte Senden (KBS) ist folgendermaßen aufgebaut:



Die Bits haben folgende Bedeutung:

Bit 1 "Auftrag läuft"

- wird vom Systemprogramm gesetzt, wenn ein Auftrag gestartet ist.
- wird vom Systemprogramm zurückgesetzt, wenn der Auftrag beendet ist.
- können Sie auswerten, wenn Sie wissen wollen, ob es sinnvoll ist einen neuen Auftrag zu starten.

Bit 2 "Auftrag fertig ohne Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag ohne Fehler abgeschlossen wurde.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag das Bit 1 "Auftrag läuft" gesetzt wird.
- können Sie auswerten, wenn Sie wissen wollen, ob der Auftrag ohne Fehler abgeschlossen wurde.

Bit 3 "Auftrag fertig mit Fehler"

- wird vom Systemprogramm gesetzt, wenn der Auftrag mit Fehler abgeschlossen wurde.
- können Sie nach erfolgter Auswertung zurücksetzen oder wird vom Systemprogramm zurückgesetzt, sobald für einen neuen Auftrag das Bit 1 "Auftrag läuft" gesetzt wird.
- können Sie auswerten, wenn Sie wissen wollen, ob der Auftrag mit Fehler abgeschlossen wurde.

Bit 6 "Schnittstelle zurücksetzen"

- können Sie setzen, wenn Sie die Schnittstelle zurücksetzen wollen.
- wird vom Systemprogramm zurückgesetzt, sobald das Systemprogramm die Schnittstelle rückgesetzt hat.

Bit 7 "SEND-Auftrag starten"

- können Sie setzen, wenn Sie einen SEND-Auftrag starten wollen
- wird vom Systemprogramm zurückgesetzt, sobald das Systemprogramm den SEND-Auftrag beendet hat.

Wenn Sie die Schnittstelle zurücksetzen, werden die Parameter der Sicherungsschicht des statischen Parametersatzes (ab DW 5) neu übernommen:

- Modus
- Priorität
- Zeichenverzugszeit
- Quittungsverzugszeit
- Aufbauversuche
- Wiederholversuche.

Haben Sie KBS-Bit 6 und KBS-Bit 7 gleichzeitig gesetzt oder im zeitlichen Abstand von ≤ 10ms bei der CPU 928, zuerst KBS-Bit 6 und dann KBS-Bit 7 gesetzt, so führt das Systemprogramm zuerst das Zurücksetzen, danach den SEND- Auftrag aus.

Haben Sie im zeitlichen Abstand von ≤ 10ms bei der CPU 928 zuerst KBS-Bit 7 und dann KBS-Bit 6 gesetzt, wird die Reihenfolge der beiden Aufträge nicht garantiert.

Mit dem Zurücksetzen brechen Sie einen laufenden Partner-SEND-Auftrag ab.

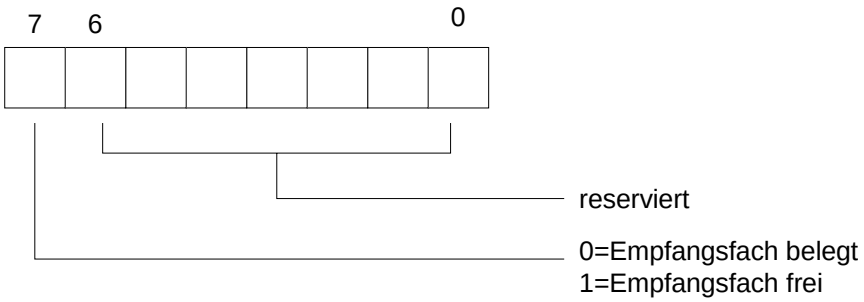
Hinweis!

Die Parameter der Bit-Übertragungsschicht der Übertragungsleitung (statischer Parametersatz bis einschließlich DW 4) können Sie nur durch einen Neustart der CPU verändern.

Ein Zurücksetzen der Schnittstelle reicht dazu nicht aus.

KBE **Koordinierungsbyte Empfangen**

Solange das KBE-Bit 7 zurückgesetzt ist, werden keine empfangenen Daten ins Empfangsfach eingetragen.



Achtung!

Sorgen Sie in Ihrem Anwenderprogramm dafür, dass Ihre empfangenen Daten möglichst schnell bearbeitet werden und geben Sie gleich im Anschluss daran das Empfangsfach über KBE-Bit 7 wieder frei.

Werden weitere Daten empfangen, während das Empfangsfach über KBE-Bit 7 gesperrt ist, stehen diese solange im Dual-Port-RAM bis Sie das Empfangsfach über KBE-Bit 7 wieder freigeben.

Es kann noch ein weiterer Empfang von Daten im Empfangspuffer des Kommunikations-Prozessors zwischengespeichert werden, solange das Empfangsfach gesperrt ist. Jeder weitere Empfang von Daten wird verworfen.

Hinweis

Bei Neustart und Spannungswiederkehr wird das KBS mit 00h (kein Auftrag läuft) und das KBE mit 80h (Empfangsfach frei) vorbelegt.

Inbetriebnahme

Voraussetzung

Ihre CPU besitzt ein Schnittstellen-Hybridmodul für die 2. Schnittstelle SI2 Sie haben das Schnittstellenmodul in die CPU eingebaut, die CPU steckt im ZG-Rahmen und die Spannung ist ausgeschaltet.

Vorgehensweise

Für die Inbetriebnahme der Datenübertragung mit der Prozedur 3964(R) empfehlen wir folgende Vorgehensweise:

Schritt	Handlung
1	Verbinden Sie die CPU mit dem Partner über eine Standard-Steckleitung.
2	Schalten Sie die Versorgungsspannung ein.
3	Übertragen Sie Ihr Anwenderprogramm zur CPU. Das Anwenderprogramm muss folgende **Datenbausteine** oder **erweiterte Datenbausteine** enthalten, die die Schnittstelle SI 2 als Datenübertragung mit der Prozedur 3964(R) parametrieren: DX 2 DB/DX mit dem statischen Parametersatz DB/DX mit dem Sendefach [1] DB/DX mit dem Empfangsfach [2] DB/DX mit dem KBS und dem KBE (nur wenn das KBS in einem DB/DX liegt) [3]
4	Führen Sie einen **Neustart** durch.
5	Bereiten Sie den Partner ebenfalls auf die Datenübertragung mit der Prozedur 3964(R) vor.

[1] Nur wenn die CPU selbst aktive Aufträge anstößt.

[2] Nur wenn die CPU selbst passive Aufträge empfängt.

[3] Falls KBS und KBE ein Merker oder S-Merker ist, ist es bereits in der CPU vorhanden und Sie brauchen es nicht zu übertragen.

Ablauf

Das Systemprogramm prüft bei Neustart oder Spannungswiederkehr das Vorhandensein von

- DX 2
- dem statischen Parametersatz
- KBS und KBE

und die korrekte Parametrierung von DX 2

Vorhanden und korrekt parametriert

- Das Systemprogramm initialisiert die Schnittstelle SI 2 für die Datenübertragung mit der Prozedur 3964(R)
- Die rote LED SI 2 erlischt, sobald die Datenübertragung mit der Prozedur 3964(R) betriebsbereit ist
- Das Systemprogramm setzt das KBS-Bit 7 zurück (d.h. kein SEND-Auftrag läuft)
- Das Systemprogramm setzt das KBE-Bit 7 (d.h. Empfangsfach freigegeben)

↓

Die Datenübertragung mit der Prozedur 3964(R) ist betriebsbereit

DX 2 vorhanden und nicht korrekt parametriert

- Die CPU bleibt in "STOP"
- Zusätzlich zur STOP-LED leuchtet die rote LED SI 2
- Das Systemprogramm gibt eine Fehlermeldung in BS 3 und BS 4 aus.

Betrieb

Voraussetzung

Die Schnittstelle SI 2 ist für die Datenübertragung mit der Prozedur 3964(R) parametriert.

Die LED SI 2 leuchtet nicht. Die Datenübertragung mit der Prozedur 3964(R) ist betriebsbereit.

Arbeitsweise

Das Systemprogramm prüft jetzt ständig, ob das Koordinierungsbyte Senden (KBS) und Empfangen (KBE) vorhanden ist. Ist es nicht vorhanden (z.B. weil der DB/DX gelöscht wurde), ruft das Systemprogramm den OB 35 auf.

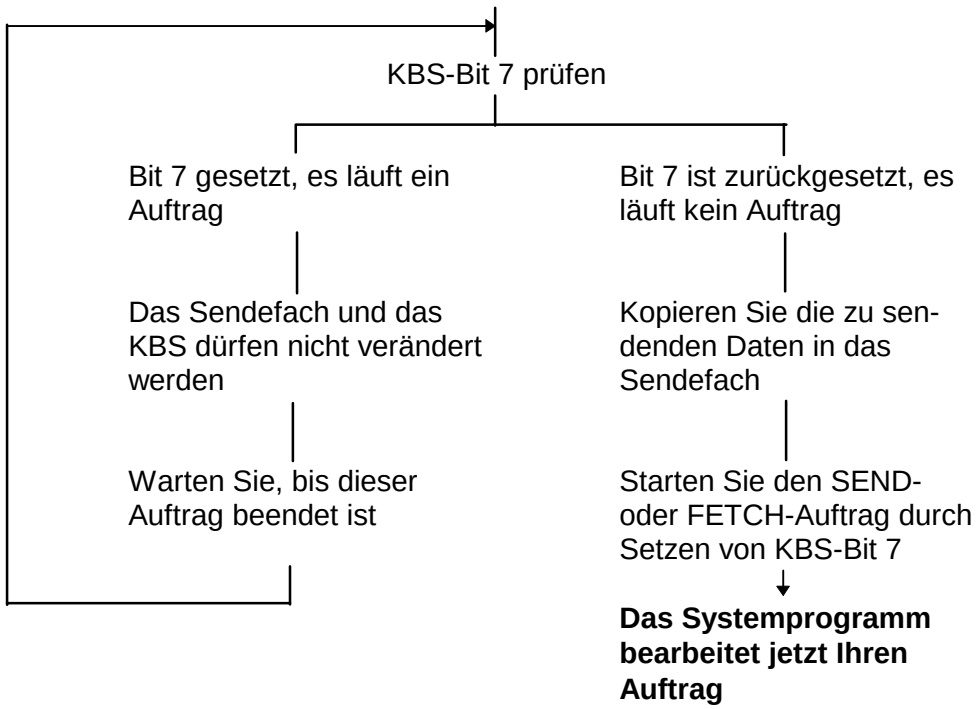
Die Fehlernummer wird in AKKU 1 eingetragen.

Das Systemprogramm bearbeitet jetzt die Aufträge, die Sie in Ihrem Anwenderprogramm programmiert haben, bzw. die Aufträge, die von Ihrem Partner angestoßen werden. Wie Sie die Aufträge in Ihrem Anwenderprogramm programmieren müssen, erfahren Sie in diesem Kapitel.

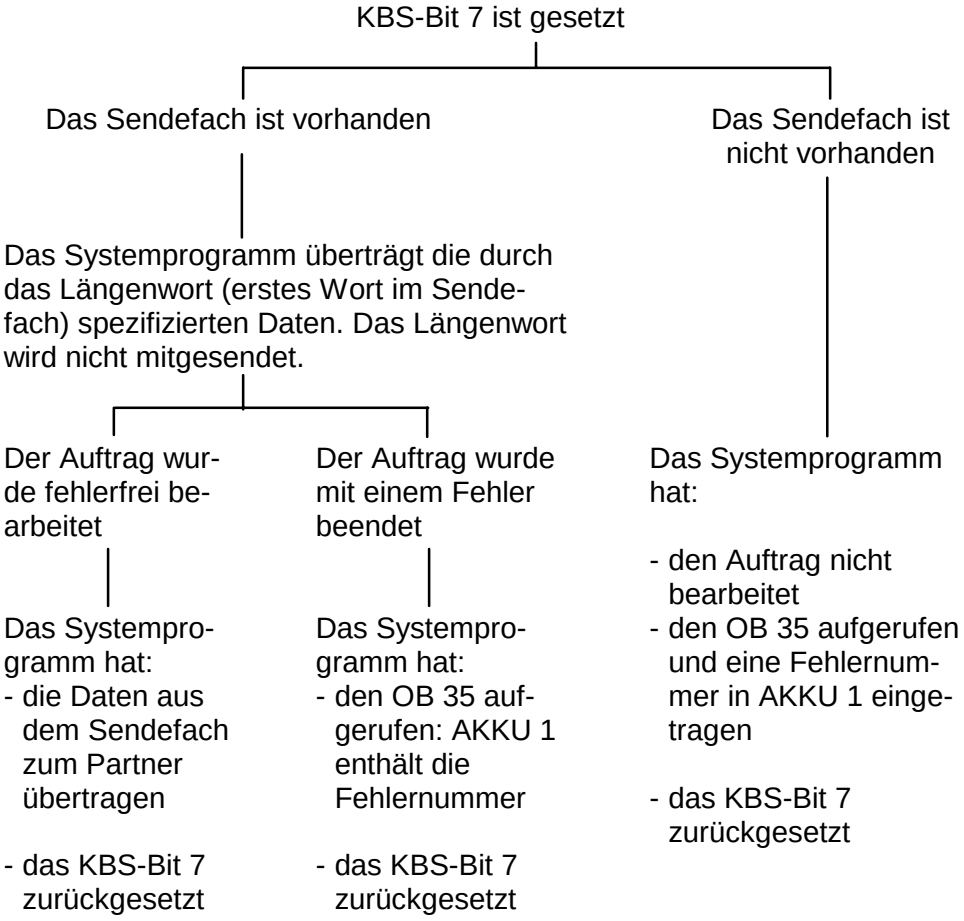
SEND-Auftrag

Wenn Sie einen SEND-Auftrag programmieren wollen, beachten Sie folgenden Ablauf.

Auftrag starten


KBS-Bit 7 prüfen
Bit 7 gesetzt, es läuft ein Auftrag
Bit 7 ist zurückgesetzt, es läuft kein Auftrag
Das Sendefach und das KBS dürfen nicht verändert werden
Kopieren Sie die zu sendenden Daten in das Sendefach
Warten Sie, bis dieser Auftrag beendet ist
Starten Sie den SEND- oder FETCH-Auftrag durch Setzen von KBS-Bit 7
Das Systemprogramm bearbeitet jetzt Ihren Auftrag

Das nachfolgende Diagramm zeigt, welche Reaktionen auftreten können, wenn das KSB-Bit 7 gesetzt ist und das Systemprogramm Ihren Auftrag bearbeitet.

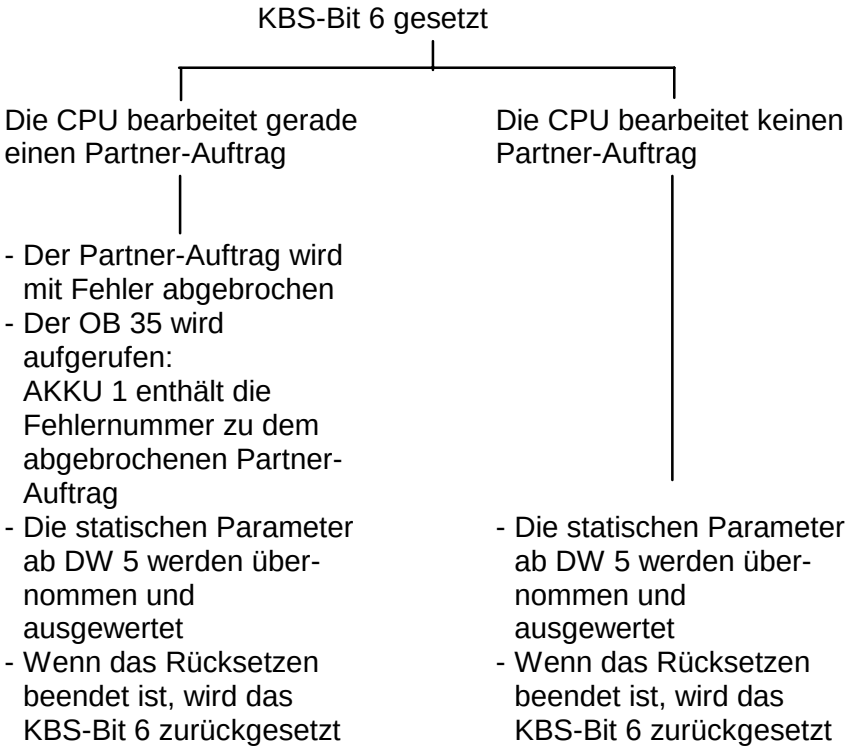

KBS-Bit 7 ist gesetzt
Das Sendefach ist vorhanden
Das Sendefach ist nicht vorhanden
Das Systemprogramm überträgt die durch das Längenwort (erstes Wort im Sendefach) spezifizierten Daten. Das Längenwort wird nicht mitgesendet.
Der Auftrag wurde fehlerfrei bearbeitet
Der Auftrag wurde mit einem Fehler beendet
Das Systemprogramm hat:
- den Auftrag nicht bearbeitet
- den OB 35 aufgerufen und eine Fehlernummer in AKKU 1 eingetragen
- das KBS-Bit 7 zurückgesetzt
Das Systemprogramm hat:
- die Daten aus dem Sendefach zum Partner übertragen
- das KBS-Bit 7 zurückgesetzt
Das Systemprogramm hat:
- den OB 35 aufgerufen: AKKU 1 enthält die Fehlernummer
- das KBS-Bit 7 zurückgesetzt

Hinweis!

Sie dürfen erst dann einen neuen Auftrag starten, wenn der vorangegangene beendet ist.

Schnittstelle rücksetzen

Sie haben das KBS-Bit 6 gesetzt und damit einen Rücksetz-Auftrag abgegeben. Das nachfolgende Diagramm zeigt die möglichen Reaktionen.


KBS-Bit 6 gesetzt
Die CPU bearbeitet gerade einen Partner-Auftrag
Die CPU bearbeitet keinen Partner-Auftrag
- Der Partner-Auftrag wird mit Fehler abgebrochen
- Der OB 35 wird aufgerufen: AKKU 1 enthält die Fehlernummer zu dem abgebrochenen Partner-Auftrag
- Die statischen Parameter ab DW 5 werden übernommen und ausgewertet
- Wenn das Rücksetzen beendet ist, wird das KBS-Bit 6 zurückgesetzt
- Die statischen Parameter ab DW 5 werden übernommen und ausgewertet
- Wenn das Rücksetzen beendet ist, wird das KBS-Bit 6 zurückgesetzt

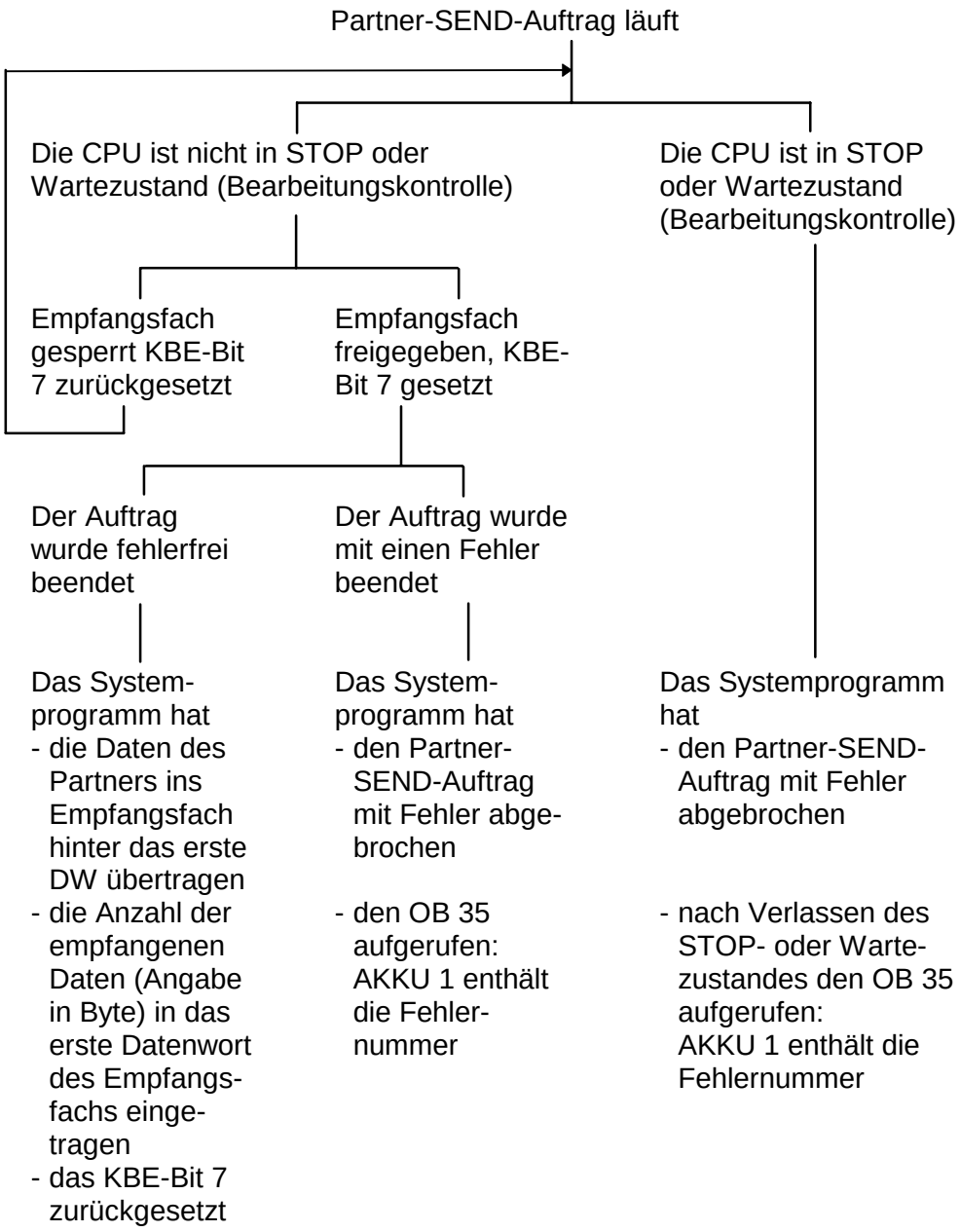
Partner-SEND-Auftrag

Partner-SEND-Aufträge werden nicht vom Anwenderprogramm sondern vom Systemprogramm selbständig im Hintergrund bearbeitet.

Das Systemprogramm setzt das KBE-Bit 7 zurück (Empfangsfach gesperrt), wenn die Daten vollständig ins Empfangsfach eingetragen wurden. Sie können nun die empfangenen Daten auswerten.

Das Empfangsfach kann erst wieder überschrieben werden, wenn Sie es über KBE-Bit 7 wieder freigegeben haben.

Das folgende Diagramm zeigt, welche Situationen nach Ablauf eines Partner-SEND-Auftrages auftreten können.


Partner-SEND-Auftrag läuft
Die CPU ist nicht in STOP oder Wartezustand (Bearbeitungskontrolle)
Die CPU ist in STOP oder Wartezustand (Bearbeitungskontrolle)
Empfangsfach gesperrt KBE-Bit 7 zurückgesetzt
Empfangsfach freigegeben, KBE-Bit 7 gesetzt
Der Auftrag wurde fehlerfrei beendet
Der Auftrag wurde mit einen Fehler beendet
Das System-programm hat
- die Daten des Partners ins Empfangsfach hinter das erste DW übertragen
- die Anzahl der empfangenen Daten (Angabe in Byte) in das erste Datenwort des Empfangs-fachs einge-tragen
- das KBE-Bit 7 zurückgesetzt
Das System-programm hat
- den Partner-SEND-Auftrag mit Fehler abge-brochen
- den OB 35 aufgerufen: AKKU 1 enthält die Fehler-nummer
Das Systemprogramm hat
- den Partner-SEND-Auftrag mit Fehler abgebrochen
- nach Verlassen des STOP- oder Warte-zustandes den OB 35 aufgerufen: AKKU 1 enthält die Fehlernummer

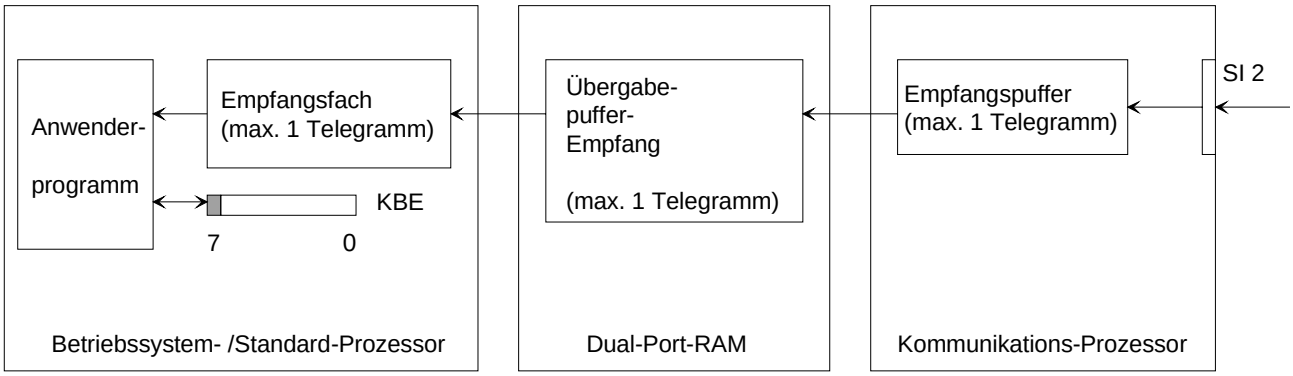
Hinweis!

Sorgen Sie in Ihrem Anwenderprogramm dafür, dass Ihre empfangenen Daten möglichst schnell bearbeitet werden und geben Sie gleich im Anschluss daran das Empfangsfach über KBE- Bit 7 wieder frei.

Besonderheiten beim Empfang

In das Empfangsfach kann immer nur ein Telegramm (max. 4096 Byte lang) eingetragen werden, wobei im Übergabepuffer-Empfang des Dual-Port-RAM und im Empfangspuffer jeweils ein weiteres Telegramm zwischengespeichert werden kann.

Wenn Sie das Empfangsfach geleert haben, und im Koordinierungsbyte Empfang (KBE) das Bit 7 (Empfangsfach frei) gesetzt haben, wird das nächste Telegramm, sofern vorhanden, vom Übergabepuffer-Empfang des Dual-Port-RAM ins Empfangsfach und abermals das nächste Telegramm, sofern vorhanden, vom Empfangspuffer in den Übergabepuffer-Empfang im Dual-Port-RAM geschoben.



Verhalten bei speziellen Betriebs-zuständen

Stoppzustand

Im Betriebszustand "STOP" findet keine Anwenderprogrammbearbeitung statt, d.h. Sie können keinen neuen Auftrag starten. Beim Übergang vom Betriebszustand "RUN" in den Betriebszustand "STOP" gilt für einen bereits laufenden, vom Anwenderprogramm angestoßenen Auftrag:

- Der Auftrag wird mit Fehler abgebrochen, wenn noch Daten zwischen dem Betriebssystem-Prozessor und dem Dual-Port-RAM transferiert werden müssen.
- Der Auftrag wird normal beendet, wenn keine Daten mehr zwischen dem Betriebssystem-Prozessor und dem Dual-Port-RAM transferiert werden müssen.

Stößt der Koppelpartner im Betriebszustand "STOP" einen Auftrag an, wird dieser negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Beim Übergang vom Betriebszustand "RUN" in den Betriebszustand "STOP" wird ein bereits laufender, vom Koppelpartner angestoßener Auftrag mit Fehler abgebrochen, d.h. negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Warte-zustand

Die CPU nimmt bei der PG-Online-Funktion "BEARBEITUNGSKON-TROLLE" den Betriebszustand "Wartezustand" ein. Im "Wartezustand" findet keine Anwenderprogrammbearbeitung statt, d.h. Sie können keinen neuen Auftrag starten. Ein bereits laufender, vom Anwenderprogramm angestoßener Auftrag wird nach Beendigung des Betriebszustandes "Wartezustand" fortgesetzt und abgeschlossen.

Stößt der Koppelpartner im Betriebszustand "Wartezustand" einen Auftrag an, wird dieser negativ quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen. Bei Eintritt in den Betriebszustand "Wartezustand" wird ein bereits laufender, vom Koppelpartner angestoßener Auftrag mit Fehler quittiert. Eine entsprechende Fehlermeldung wird in den Fehlerbereich der Schnittstelle eingetragen.

Prozeduren

Die Prozeduren 3964 und 3964R gestatten eine unkomplizierte Datenübertragung im Quasi-Vollduplexbetrieb mit guter Übertragungsqualität. Partner können außer der CPU 928 noch Fremdgeräte und Fremdrechner sein.

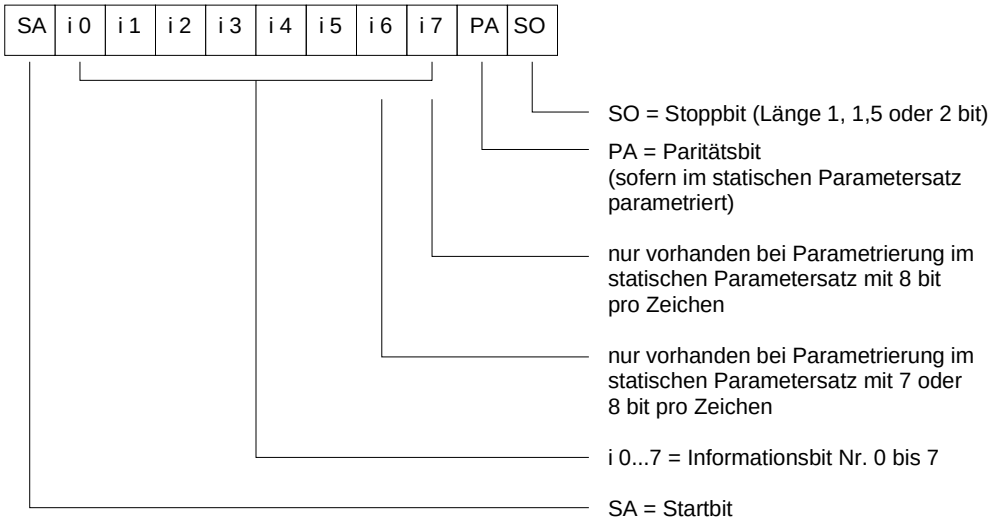
Steuerzeichen

Beide Prozeduren steuern den Datenfluss zwischen Ihrer CPU 928 und dem Partner. Sie sind **asynchrone, bitserielle Übertragungsverfahren**.

Sende- und Empfangsparameter (Baudrate, Parity,...) müssen Sie in der CPU 928 und im Partner gleich einstellen, die Priorität müssen Sie in der CPU 928 und im Partner verschieden einstellen.

Beide Prozeduren fügen bei der Datenübertragung den Informationszeichen **Steuerzeichen** hinzu. Durch diese Steuerzeichen kann der Partner kontrollieren, ob die Daten vollständig und fehlerfrei bei ihm angekommen sind.

Die CPU 928 bzw. der Partner sendet die Zeichen bitseriell wie folgt:


SA
i 0
i 1
i 2
i 3
i 4
i 5
i 6
i 7
PA
SO
SO = Stoppbit (Länge 1, 1,5 oder 2 bit)
PA = Paritätsbit
(sofern im statischen Parametersatz parametriert)
nur vorhanden bei Parametrierung im statischen Parametersatz mit 8 bit pro Zeichen
nur vorhanden bei Parametrierung im statischen Parametersatz mit 7 oder 8 bit pro Zeichen
i 0...7 = Informationsbit Nr. 0 bis 7
SA = Startbit

Die Steuerzeichen für die Prozeduren sind der DIN-Norm 66003 für den 7-Bit-Code entnommen. Übertragen wird jedoch mit der Zeichenlänge 6,7 oder 8 Bit.

Wird als Informationszeichen das Zeichen DLE übertragen, so wird dieses auf der Sendeleitung doppelt gesendet zur Unterscheidung vom Steuerzeichen DLE beim Verbindungsauf- und Verbindungsabbau (DLE-Verdopplung). Der Empfänger macht die DLE-Verdopplung wieder rückgängig.

Für die anderen Informationszeichen ist kein Code vorgeschrieben.

Senden

Zum Aufbau der Verbindung sendet die Prozedur das Steuerzeichen STX. Antwortet das Partnergerät vor Ablauf der Quittungsverzugszeit (QVZ) mit dem Zeichen DLE, geht die Prozedur in den Sendebetrieb über. Antwortet das Partnergerät mit NAK (Nachricht unbrauchbar), einem beliebigen anderen Zeichen (außer DLE) oder die QVZ verstreicht ohne Reaktion, wiederholt die Prozedur den Verbindungsaufbau. Nach der im statischen Parametersatz festgelegten Anzahl vergeblicher Aufbauversuche bricht die Prozedur den Verbindungsaufbau ab und sendet das Zeichen NAK an das Partnergerät.

Das Systemprogramm ruft den OB 35 auf und meldet den Fehler im Verbindungsaufbau (Fehlernummer im AKKU 1).

Gelingt der Verbindungsaufbau, werden die im aktuellen Ausgabepuffer enthaltenen Nutzdaten mit den gewählten Übertragungsparametern an den Partner gesendet. Dieser überwacht den zeitlichen Abstand der ankommenden Zeichen. Der Abstand zwischen zwei Zeichen darf nicht mehr als die Zeichenverzugzeit (ZVZ) betragen.

DLE-Verdopplung

Jedes im Puffer vorgefundene Zeichen DLE wird als zwei Zeichen DLE gesendet (DLE-Verdopplung), d.h. jedes Datum /10h/ wird zweimal gesendet. Der Empfänger macht die DLE-Verdopplung wieder rückgängig.

Blockprüfsumme

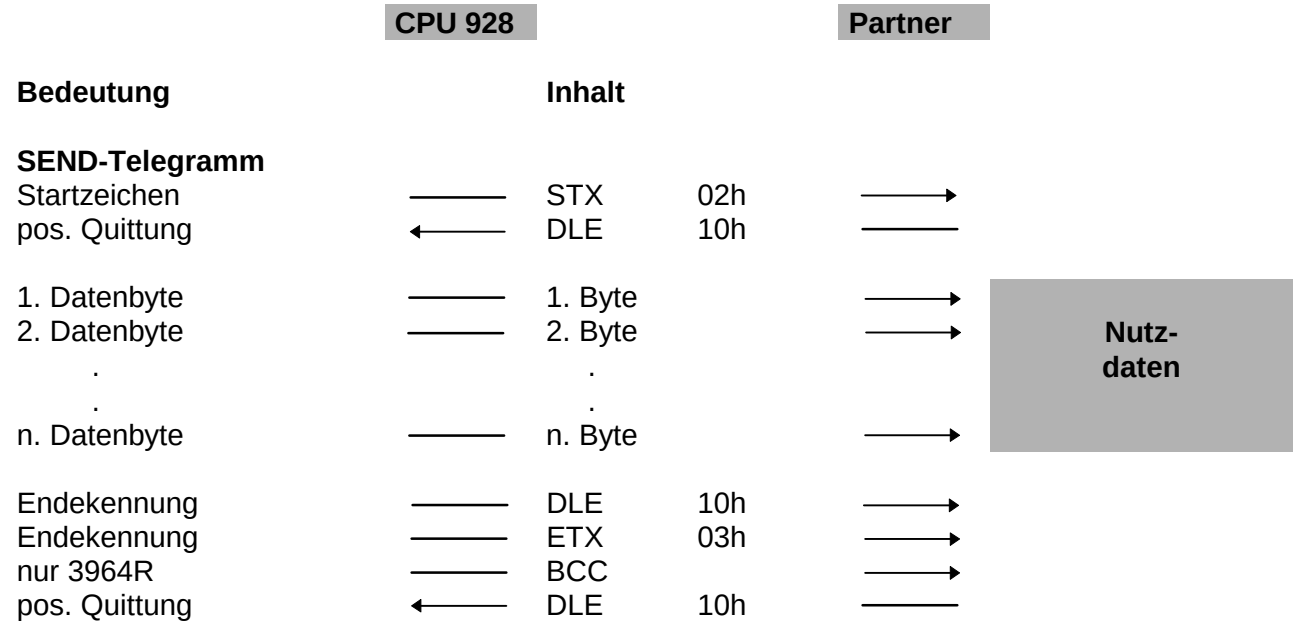
Nach dem Senden des Pufferinhalts fügt die Prozedur die Zeichen DLE, ETX und **nur bei 3964R** die Blockprüfsumme BCC als Endekennung an und wartet auf ein Quittungszeichen. Das Blockprüfzeichen BCC ist ein Byte. Es wird gebildet aus der geraden Längsparität aller Zeichen (EXOR-Verknüpfung), beginnend mit dem ersten Zeichen nach dem Verbindungsaufbau und endet nach den Zeichen DLE ETX des Verbindungsabbaus. Die Prüfsumme wird nach der DLE-Verdopplung gebildet.

Sendet das Partnergerät innerhalb der QVZ das Zeichen DLE, wurde der Datenblock fehlerfrei übernommen. Antwortet das Partnergerät mit NAK, einem beliebigen anderen Zeichen (außer DLE), einem gestörten Zeichen oder die QVZ verstreicht ohne Reaktion, beginnt die Prozedur das Senden erneut mit dem Verbindungsaufbau STX. Nach Ablauf der im statischen Parametersatz spezifizierten Anzahl von Versuchen (Wiederholanzahl), den Datenblock zu senden, bricht die Prozedur das Verfahren ab und sendet NAK an das Partnergerät.

Das Systemprogramm ruft den OB 35 auf und meldet den Fehler (Fehlernummer in AKKU 1).

Sendet das Partnergerät während einer laufenden Sendung das Zeichen NAK, beendet die Prozedur den Block und wiederholt ihn in der oben beschriebenen Weise. Bei einem anderen Zeichen wartet die Prozedur zunächst auf den Ablauf der ZVZ und sendet anschließend NAK, um das Partnergerät in den Ruhestand zu bringen. Danach beginnt die Prozedur das Senden erneut mit dem Verbindungsaufbau STX.

Beispiel für einen fehlerlosen Datenverkehr beim Senden



Empfangen

Im Ruhezustand, wenn kein Sendeauftrag zu bearbeiten ist, wartet die Prozedur auf den Aufbau der Verbindung durch den Partner.

Empfängt die Prozedur im Ruhestand ein beliebiges Zeichen (außer STX oder NAK), wartet sie auf den Ablauf der Zeichenverzugszeit (ZVZ) und sendet dann ein Zeichen NAK. Das Systemprogramm ruft OB 35 auf und meldet den Fehler (Fehlernummer in AKKU 1).

Empfangspuffer

Empfängt die Prozedur das Zeichen STX und steht ihr ein leerer Empfangspuffer zur Verfügung, antwortet sie mit DLE. Ankommende Empfangszeichen werden nun im Empfangspuffer abgelegt. Werden zwei aufeinanderfolgende Zeichen DLE empfangen, wird nur ein Zeichen DLE in den Empfangspuffer übernommen. Ist der Empfangspuffer voll, bevor das Partnergerät den Verbindungsabbau eingeleitet hat, wird die Verbindung mit NAK abgebrochen.

Nach jedem Empfangszeichen wird während der Zeichenverzugszeit (ZVZ) auf das nächste Zeichen gewartet. Verstreicht die ZVZ ohne Empfang, wird das Zeichen NAK an das Partnergerät gesendet. Das Systemprogramm ruft OB 35 auf und meldet den Fehler (Fehlernummer in AKKU 1).

Empfang fehlerfrei / fehlerhaft

Erkennt die Prozedur **3964** die Zeichenfolge DLE ETX, beendet sie den Empfang und sendet DLE für einen fehlerfrei empfangenen oder NAK für einen fehlerhaft empfangenen Block an das Partnergerät.

Die Prozedur kehrt zurück in den Ruhestand.

Erkennt die Prozedur **3964R** die Zeichenfolge DLE ETX BCC, beendet sie den Empfang. Sie vergleicht das empfangene Blockprüfzeichen BCC mit

der intern gebildeten Längsparität. Ist das Blockprüfzeichen korrekt und kein anderer Empfangsfehler aufgetreten, sendet die Prozedur 3964R DLE und kehrt zurück in den Ruhestand.

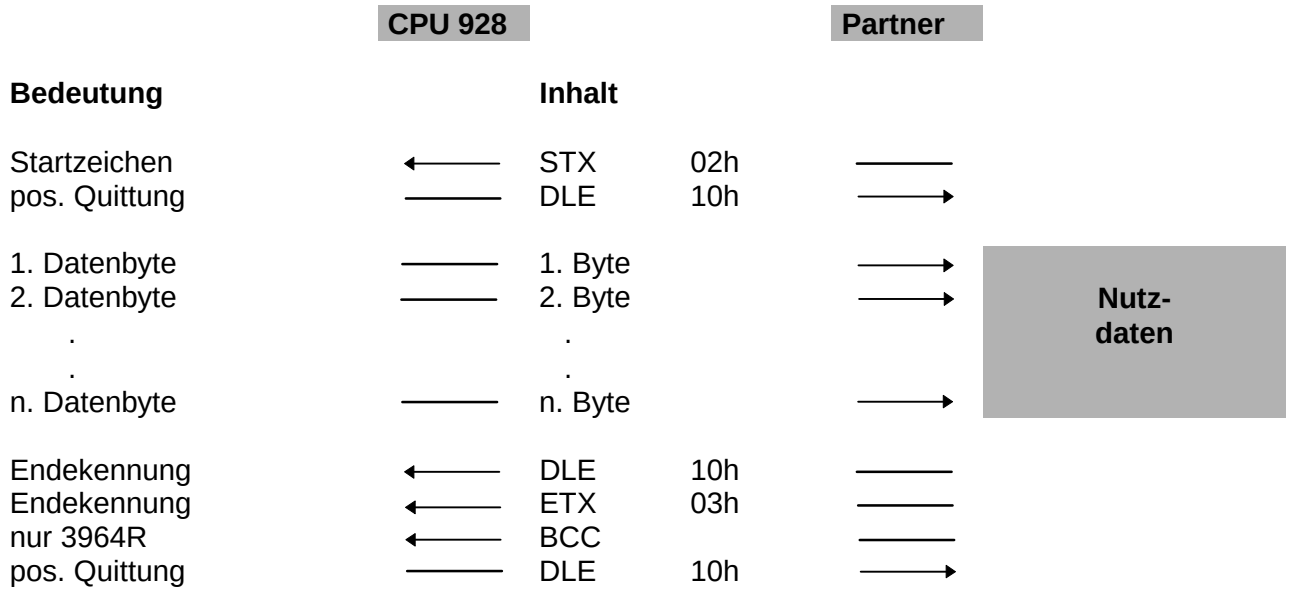
Bei fehlerhaftem BCC oder einem anderen Empfangsfehler wird NAK an das Partnergerät gesendet. Anschließend wird eine Wiederholung erwartet.

Steht beim Verbindungsaufbau mit STX kein leerer Empfangspuffer zur Verfügung, wird eine Wartezeit von 400 ms gestartet. Liegt nach dieser Zeit noch kein leerer Empfangspuffer vor, ruft das Systemprogramm OB 35 auf und meldet den Fehler (Fehlernummer in AKKU 1).

Die Prozedur sendet ein Zeichen NAK und geht wieder in den Ruhestand zurück. Andernfalls sendet die Prozedur das Zeichen DLE und empfängt die Daten in der oben beschriebenen Weise.

Treten während des Empfangens Übertragungsfehler auf (verlorenes Zeichen, Rahmenfehler, Paritätsfehler usw.), so wird bis zum Verbindungsabbau weiterempfangen und dann NAK an das Partnergerät gesendet. Anschließend wird eine Wiederholung erwartet. Kann der Block auch nach der im statischen Parametersatz spezifizierten Anzahl von Wiederholversuchen nicht fehlerfrei empfangen werden oder wird die Wiederholung vom Partnergerät nicht innerhalb einer Wartezeit von 4 s gestartet, bricht die Prozedur den Empfang ab. Das Systemprogramm ruft OB 35 auf und meldet den Fehler (mit Fehlernummer in AKKU 1).

Beispiel für einen fehlerlosen Datenverkehr beim Empfang



Hinweis!

Die Prozedur 3964(R) sendet nach der Initialisierung oder dem Zurücksetzen das Zeichen NAK an den Partner, um das Partnergerät in den Ruhezustand zu bringen.

Initialisierungs-konflikt

Antwortet ein Gerät auf den Sendewunsch (Zeichen STX) seines Partners innerhalb der Quittungsverzugszeit (QVZ) nicht mit der Quittung DLE oder NAK, sondern ebenfalls mit dem Zeichen STX, so liegt ein Initialisierungskonflikt vor. Beide Geräte möchten einen vorliegenden Sendeauftrag ausführen. Das Gerät mit der niedrigeren Priorität stellt seinen Sendeauftrag zurück und antwortet mit dem Zeichen DLE. Das Gerät mit der höheren Priorität sendet seine Daten in der zuvor beschriebenen Weise. Nach dem Verbindungsabbau kann das Gerät mit der niedrigeren Priorität seinen Sendeauftrag ausführen.

Priorität

Für den Fall des Initialisierungskonflikts können Sie die Priorität der CPU 928 einstellen. Beim Partner stellen Sie die entgegengesetzte Priorität ein.

Der Parameter Priorität wird im statischen Parametersatz eingestellt.

Beispiel zur Lösung des Initialisierungskonfliktes:

	CPU 928 niedrige Priorität			**Partner** höhere Priorität	
Bedeutung		**Inhalt**			
Startzeichen	———	STX	02h	⟶	
Startzeichen	⟵	STX	02h	———	
pos. Quittung	———	DLE	10h	⟶	
1. Datenbyte	⟵	1. Byte		———	**Nutzdaten**
2. Datenbyte	⟵	2. Byte		———	
.		.			
.		.			
n. Datenbyte	⟵	n. Byte		———	
Endekennung	⟵	DLE	10h	———	
Endekennung	⟵	ETX	03h	———	
nur 3964R	⟵	BCC		———	
pos. Quittung	———	DLE	10h	⟶	
Startzeichen	———	STX	02h	⟶	
pos. Quittung	⟵	DLE	10h	———	
	.			.	
	.			.	
	.			.	

Prozedurfehler

Die Prozedur erkennt Fehler, die durch ein fehlerhaftes Verhalten des Partnergerätes ausgelöst werden und Fehler, die durch Störungen auf der Leitung verursacht werden.

In beiden Fällen wird zunächst versucht, beim Wiederholen den Datenblock richtig zu senden bzw. zu empfangen. Kann der Datenblock bis zur Maximalanzahl der Wiederholungen nicht fehlerfrei gesendet oder empfangen werden, bricht die Prozedur das Senden bzw. Empfangen ab.

Sie meldet die Fehlernummer für den ersten erkannten Fehler und begibt sich in den Ruhezustand. Das Systemprogramm ruft den OB 35 auf, die Fehlernummer steht in AKKU 1.

Störeinflüsse

Falls das Systemprogramm des öfteren mit Fehlernummer in AKKU 1 für Sende- und Empfangswiederholungen den OB 35 aufruft, sollten Sie auf gelegentliche Störungen des Datenverkehrs schließen.

Dies wird durch die Vielzahl der Wiederholungen jedoch ausgeglichen. In diesem Fall empfehlen wir Ihnen, die Übertragungsstrecke auf Störeinflüsse zu untersuchen, da die Nutzdatenrate und Sicherheit der Übertragungen bei vielen Wiederholungen sinkt. Die Ursache der Störung kann jedoch auch in einem fehlerhaften Verhalten des Partners liegen.

Hinweis!

Bei BREAK auf der Empfangsleitung wird vom Systemprogramm im OB 35 (AKKU 1) der BREAK-Zustand gemeldet. Es wird keine Wiederholung gestartet. Dieser Zustand wird solange anstehen, bis der BREAK-Zustand beseitigt ist. Der OB 35 wird erst wieder aufgerufen, wenn ein weiterer Fehler aufgetreten ist.

Für alle erkannten Übertragungsfehler (verlorenes Zeichen, Rahmenfehler, Paritätsfehler) wird eine einheitliche Nummer gemeldet, gleichgültig, ob der Fehler beim Senden oder Empfangen eines Datenblocks erkannt wurde. Der Fehler wird jedoch nur gemeldet, wenn die Wiederholungen zuvor erfolglos verliefen. Wird das gestörte Zeichen im Ruhezustand der Prozedur empfangen, ruft das Systemprogramm OB 35 und meldet den Fehler (Fehlernummer in AKKU 1), um Sie auf schwere Störeinflüsse auf der Übertragungsstrecke hinzuweisen.

A Anhang

Gesamtindex

Die Bände sind wie folgt gegliedert:

Band 1:	Teil 1 ... Teil 11	Hardware und Programmierung
Band 2:	Teil 12 ... Teil 15	Integrierte Funktionen und Kommunikation

2

20mA-Schnittstelle 2-11

3

3964(R) 15-1
- Betrieb 15-24
- Inbetriebnahme 15-22
- Parametrierung 15-9
- Prozeduren 15-30

A

Adressierungsarten 1-9
AKKU 4-13
- Inhalte rollen 12-8
- löschen 12-7

Alarm
- gemeinsam verzögern 12-15
- Prozess- 5-23
- Reaktion 4-11
- sperren 12-10

Analogwertbausteine 13-32
Anlauf 4-10; 5-11
- Anwenderschnittstellen 5-13
- -fehler 6-24
- -unterbrechung 5-14
- -verhalten 2-17

A-NR 13-4
Anwenderkontrollpunkte 11-3
Anwenderprogramm 3-1
- bearbeiten 4-10

Anwendungsbereiche 1-5
Anzeigenwort 8-16
Anzeigenwort 13-10
ANZW 13-4
Arbeitsweise einer CPU 1-6
Aufbau CPU 928 2-3
Aufruf im STOP 11-8

B

BA/BB-Bereich 8-9
BASP 5-4
Baustein
- Ablage im Speicher 3-10
- -adressliste im DB 0 8-7
- -köpfe 8-6
- löschen 11-6
- -schachtelung 4-6
- -Stack lesen 12-45
- übertragen 11-6

Betriebszustände 5-1; 5-2
- Anlauf 5-11
- RUN 5-15
- STOP 5-8

Bit
- -anzeigen 4-15
- -belegung BS 8-12

BLGR 13-5
Blockgröße 13-31
BS/BT-Bereich 8-9

C

CONTROL 13-26
CPU 928-Varianten 1-4

D

Darstellungsarten
- FUP, KOP, AWL 3-2

Datenbaustein 3-29
- aufschlagen 3-31
- Beispiele 3-32
- Datenblock kopieren 12-56
- erstellen 3-30
- Merker übertragen 12-62; 12-65
- spezielle 3-34
- testen 12-54
- variabler Zugriff 12-50
- verschieben, duplizieren 12-98

DB/DX testen 12-54
DBNR 13-5
DB-Zugriff variabel 12-50
Digitalverknüpfungen 4-48
DX 0 7-2
  Aufbau 7-3
  Parameter 7-5

E

EPROM-Betrieb 2-15; 4-8
Ergebnisanzeigen 4-14

F

Fehler
  Abbruch 6-48
  Adressier- 6-43
  ANLAUF 6-24
  -anzeigen 5-3
    im Anzeigenwort 13-11
  Befehlscode- 6-32
  -behandlung 6-2
  -behandlung über OBs 6-21
  DB 0 6-25
  DB 1 6-26
  DX0 6-27
  -informationen 6-3
  Laufzeit- 6-35
  Memory-Card 6-29
  Prozedur- 15-35
  Quittungsverzug 6-43
  RUN 6-30
  SI 1, SI 2 2-8
  Weck- 6-47
  Zykluszeit- 6-46
FETCH 13-25
Frontansicht
  CPU 928-3UB12 2-6
  CPU 928-3UB21 2-5
  CPU 928-3UC11 2-4
Funktionsbausteine 3-19
  Aufbau 3-20
  Aufrufen und Parametrieren ... 3-24
  Programmierung 3-22
  Spezialbausteine 3-28

G

Grundlagen 1-1

H

Haltepunkte 11-8
Hardwarebeschreibung 2-1

I

Inbetriebnahme 2-16

K

Kachel
  Belegung 12-76
  lesen 12-74
  schreiben 12-72
  -zugriff 12-70
Kontrollpunkte 11-16

L

L- und T-Operationen .. 4-18; 4-51
Längen-Wort 13-15
Leistungsmerkmale 1-5
Leuchtdioden 2-7

M

Memory-Card 5-10

O

OB 110 12-5
OB 111 12-7
OB 112/113 12-8
OB 120 12-10
OB 121 12-13
OB 122 12-15
OB 123 12-18
OB 134 12-20
OB 135 12-20
OB 136 12-21
OB 139 12-21
OB 150 12-22
OB 151 12-29
OB 152 12-36
OB 160 bis 163 12-43
OB 170 12-45
OB 180 12-50
OB 181 12-54
OB 182 12-56
OB 186 12-60
OB 190/192 12-62
OB 191/193 12-65
OB 216 12-72
OB 216 bis 218 12-70
OB 217 12-74
OB 218 12-76
OB 220 12-78
OB 221 12-79
OB 222 12-80
OB 228 12-81
OB 230 bis 237 12-83
OB 250 12-91
OB 250/251 12-84
OB 251 12-92
OB 254/255 12-98
OBs für Anwender 3-15

Operanden der CPU 928 1-7
Operationen 4-13
Arithmetische.....4-28
Bearbeitungs- 4-62
ergänzende.....4-47
Grund- 4-16
mit Basisadressregister 9-22
organisatorische 4-54
Schiebe- 4-56
Semaphor-.....4-13
Speicher- 4-17; 4-34
Sprung-.....4-54
Umwandlungs-.....4-58
Zeit- und Zähl- 4-23; 4-49
Organisationsbausteine 3-13

P

PAE/PAA 4-12
PAFE 13-5; 13-16
Parallelbetrieb.....11-17
Ablauf 11-21
Betrieb 11-19
Inbetriebnahme.....11-19
Peripherie ansprechen.....4-22
PG-Funktionen.....11-2
PID-Regler 12-84
Abkürzungen 12-96
bearbeiten.....12-92
Funktionsbeschreibung 12-84
initialisieren.....12-91
Programmbearbeitung 4-1
Ebenen 5-5
Prinzip.....4-2
zeitgesteuerte 5-19
zyklische 5-17
Programme einer CPU.....1-6
Programmierung
Beispiele FUP, KOP, AWL 4-31
Organisationsbaustein.....3-13
Programmbausteine 3-13
Programmiersprache.....3-2
Programmorganisation 4-3
Programmtest.....11-7
Schrittbaustein.....3-13
strukturierte 3-4
Vorgehensweise 1-10

Q

QANF/ZANF 13-5
QLAE/ZLAE 13-5
QTYP/ZTYP.....13-4
QTYP/ZTYP-Parameter 13-17

R

RAM-Betrieb 4-8
RECEIVE 13-22
RECEIVE-ALL.....13-30
Rechenoperationen.....4-52
RESET 13-27
RK512.....14-1
Aufträge..... 14-4
Betrieb 14-30
Datenaustausch 14-38
Inbetriebnahme 14-28
Koordinierungsmerker..... 14-26
Parametrierung 14-10
Telegrammbeschreibung 14-43
RS232C-Schnittstelle 2-13
RS422-Schnittstelle.....
RUN 5-15

S

Schachteltiefe 4-6
Schalter.....2-9
Schrittbausteine 3-13
SEND.....13-19
SEND-ALL 13-29
Serielle Kopplung 11-17
SI 1 2-10
SI 2 2-11
Sicherheitshinweis.....1-2
Sonderfunktionen.....12-1
Akku-Schreibweise..... 12-3
Fehlerbearbeitung 12-3
OBs für Standard-FBs..... 12-83
Pseudobefehlsgrenzen..... 12-4
Schnittstellen 12-3
Übersicht 3-18; 12-1
Speicher
-belegung 8-3
komprimieren 11-5; 12-60
-zugriff
Beispiele 9-13
über absol. Adressen..... 9-1
über AKKU 1 9-6
Speicherblöcke transferieren.9-15
Speicheroperationen 4-48
Speicherorganisation 8-5
CPU 928..... 8-5
Speicherung von PBs u. DBs ..4-8
SSNR.....13-4
Standard-HTBs 13-1
Parametrierung 13-6
indirekte 13-9
Status
Baustein 11-7
Information lesen..... 12-81
Steuerbits.....6-8
STOP 5-8
SYNCHRON.....13-28
Systemkontrollpunkte.....11-3
Systemzeit stellen/lesen.....12-22

T

Transferoperationen.....9-23

U

Unterbrechungs
-behandlung.....6-2
-ereignisse.....4-12
-stack.....6-7
Unterschiede CPUs 1-12
Urlöschen.....5-9; 11-5
USTACK
Beispiele.....6-18
Inhalt.....6-13

V

VKE 3-5
VKE-Bildung 4-17
Vorzeichenerweiterung 12-78

W

WARTEZUSTAND 11-4
Weckalarm.....5-20
einzeln verzögern 12-18
sperren 12-13
uhrzeitgesteuert..... 5-20
stellen/lesen 12-29
Weckfehler.....5-22
Wortanzeigen.....4-15

Z

Zahlendarstellung.....3-6
Zählschleifen.....12-43
Zugriff auf Anzeigenbyte 12-5
Zyklus 4-10
-statistik 12-36
-überwachung
einstellen..... 12-79
neu starten..... 12-80
-zeit..... 4-11
-zeitüberwachung..... 4-12

M.Stich